



Titre: Architecture et implantation en temps réel d'un système autonome de stationnement d'un véhicule automobile
Title:

Auteur: Stéphane Gill
Author:

Date: 1995

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Gill, S. (1995). Architecture et implantation en temps réel d'un système autonome de stationnement d'un véhicule automobile [Mémoire de maîtrise, École Polytechnique de Montréal]. PolyPublie. <https://publications.polymtl.ca/6665/>
Citation:

 **Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/6665/>
PolyPublie URL:

Directeurs de recherche:
Advisors:

Programme: Non spécifié
Program:

UNIVERSITÉ DE MONTRÉAL

ARCHITECTURE ET IMPLANTATION EN TEMPS RÉEL D'UN
SYSTÈME AUTONOME DE STATIONNEMENT D'UN VÉHICULE
AUTOMOBILE

par

Stéphane GILL

DÉPARTEMENT DE GÉNIE ÉLECTRIQUE
ET DE GÉNIE INFORMATIQUE
ÉCOLE POLYTECHNIQUE

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLOME DE MAÎTRE ÈS SCIENCES APPLIQUÉES (M.Sc.A.)
(GÉNIE ÉLECTRIQUE ET GÉNIE INFORMATIQUE)
JUILLET 1995



National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-33138-5

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE

Ce mémoire intitulé:

ARCHITECTURE ET IMPLANTATION EN TEMPS RÉEL D'UN SYSTÈME
AUTONOME DE STATIONNEMENT D'UN VÉHICULE AUTOMOBILE

présenté par: Stéphane GILL

en vue de l'obtention du diplôme de: Maître ès Sciences Appliquées

a été dûment accepté par le jury d'examen constitué de :

DESANTIS Romano M., Ph.D., président

HURTEAU Richard, D.ing., directeur de recherche

COHEN Paul, Ph.D., co-directeur de recherche

HERVÉ Jean-Yves, Ph.D.

REMERCIEMENTS

.Je tiens à remercier mes parents.

SOMMAIRE

Un système de guidage autonome doit être en mesure de planifier une tâche et de l'exécuter à partir de la connaissance de son environnement. L'implantation d'un système autonome pour le stationnement d'un véhicule automobile dans un environnement urbain nécessite la résolution de différents problèmes: la perception de l'environnement 3D, la localisation du véhicule par rapport à son environnement, la planification de chemin et, l'exécution du chemin et le contrôle de position.

Dans notre réalisation, l'identification de l'espace de stationnement libre se fait à partir de données télémétriques sur l'environnement. Une représentation tridimensionnelle du profil de stationnement est calculée à partir d'une séquence d'images de l'environnement global. Une stratégie d'appariement, utilisant la phase de la transformée de Fourier du signal image, procure une carte de disparité de densité élevée pour chaque paire d'images consécutives. Une procédure non-linéaire d'estimation de la structure et du mouvement est alors utilisée pour générer des cartes locales de profondeur. Ces cartes locales sont intégrées en une représentation étendue de l'environnement, dans laquelle il devient possible d'identifier l'espace de stationnement libre.

La localisation du véhicule par rapport à son environnement se fait en mettant en correspondance les cartes locales obtenues avec le modèle global du profil de stationnement. Pour la planification de chemin deux approches ont été étudiées une approche utilisant la décomposition en cellules de l'espace de configuration ainsi qu'une approche utilisant les chemins canoniques. Pour permettre le suivi de chemin, des contrôleurs de mouvement ont été développés à partir d'un modèle dynamique simplifié du véhicule.

Pour fins d'expérimentation, un véhicule de modèle réduit a été développé. Ce véhicule de type automobile, d'échelle de 1:5, peut atteindre une vitesse de pointe de 3 m/s.

ABSTRACT

An autonomous guiding system must be able to plan a task and execute it using its knowledge of the environment. The implementation of an autonomous parking system for an automobile vehicle in an urban environment implies the resolution of different problems: 3D perception of the environment, vehicle localization with respect to the environment, path planning, and path following and position control.

In our implementation, range data are used for the identification of a free parking place in the environment. A 3D representation of the parking profil is computed from a sequence of images of the global environment. A matching strategy using the phase of the image's Fourier transform produces a high density disparity map for each consecutive pair of images. A non-linear structure and motion estimation procedure is then used to generate the local depth maps. These local maps are integrated into an extended representation of the environment, in which a free parking place is detected.

Localization of the vehicle in the environment is accomplished by matching the local depth maps with the global parking profil model. Two alternative methods have been studied for path planning: the cell decomposition method, and the canonical path method. For the path following task, motion controllers have been developed based on a simplified dynamic model of the vehicle.

For experimentation, a reduced-size (scale 1:5) vehicle has been developed. This automobile-type vehicle can reach a maximum speed of 3 m/s.

TABLE DES MATIERES

REMERCIEMENTS	iv
SOMMAIRE	v
ABSTRACT	vii
TABLE DES MATIERES	x
LISTE DES FIGURES	xii
LISTE DES TABLEAUX	xiii
LISTE DES SYMBOLES	xiv
CHAPITRE 1 Introduction	1
CHAPITRE 2 Caractérisation et modélisation	4
2.1 Introduction	4
2.2 Conception du véhicule	4
2.3 Odométrie	7
2.4 Modèle dynamique du véhicule	10
2.5 Problème de suivi de chemin	14
2.6 Conclusion	17

CHAPITRE 3 Contrôle de mouvement	18
3.1 Introduction	18
3.2 Filtrage des données odométriques	18
3.3 Contrôleur de position et de vitesse pour le système en trans- lation	21
3.4 Correction d'erreur de suivi de chemin	24
3.5 Suivi de chemin	28
3.6 Conclusion	30
 CHAPITRE 4 Planification de chemin	 32
4.1 Introduction	32
4.2 Méthode de Barraquant et Latombe [9]	33
4.3 Méthode de Laumond et Taix [11]	34
4.3.1 Chemin holonome	35
4.3.2 Chemin canonique	35
4.3.3 Chemin non-holonome	37
4.4 Conclusion	39
 CHAPITRE 5 Description du système de vision	 41
5.1 Introduction	41

5.2 Calibration des paramètres de la caméra	41
5.3 Appariement	45
5.4 Estimation de la structure	48
5.5 Conclusion	50
 CHAPITRE 6 Identification de l'espace de stationnement	 53
6.1 Introduction	53
6.2 Construction de la carte globale	53
6.3 Identification de l'espace de stationnement	56
6.4 Rétro action visuelle	60
6.5 Conclusion	61
 CHAPITRE 7 Conclusion	 63
 BIBLIOGRAPHIE	 65

LISTE DES FIGURES

CHAPITRE 1	1
1.1 Module informatique du système de stationnement	2
1.2 Scène	3
 CHAPITRE 2	 4
2.1 VIGAR	5
2.2 Architecture matériel du système informatique	7
2.3 Déplacement élémentaire	8
2.4 Représentation du véhicule dans le plan XY	11
2.5 Modèle dynamique	13
2.6 Référentiel du chemin	15
 CHAPITRE 3	 18
3.1 Estimateur d'état	19
3.2 Vitesse avant filtrage et vitesse filtrée	21
3.3 Contrôleur du système en translation	22
3.4 Consigne de vitesse et de position	23
3.5 Déplacement de 2m	24
3.6 Déplacement de 0.1m	24
3.7 Contrôleur tout-ou-rien	26
3.8 Correction d'une erreur latérale	27
3.9 Correction d'une erreur latérale et d'orientation	27
3.10 Suivi d'un arc de cercle	30

3.11 Suivi d'un créneau	30
CHAPITRE 4	32
4.1 Résultat du planificateur	33
4.2 Chemin holonome	35
4.3 Exemple de chemin canonique $G_t^+ D_u^- G_v^+$	38
4.4 Méthode de Taix et Laumond	39
CHAPITRE 5	41
5.1 Système de vision	42
5.2 Système de coordonnée de la caméra	43
5.3 Montage pour la calibration	44
5.4 Structure de calcul	46
5.5 Résultat de l'appariement	49
5.6 Résultat de l'estimation de la structure	51
CHAPITRE 6	53
6.1 Scène	54
6.2 Référentiel de la caméra	54
6.3 Carte globale	55
6.4 Identification de l'espace de stationnement	56
6.5 Exemple d'identification de l'espace de stationnement	59
6.6 Étape de la relocalisation	61
6.7 Profil local et modèle	62
CHAPITRE 7	63

LISTE DES TABLEAUX

CHAPITRE 1	1
CHAPITRE 2	4
2.1 Composants électro-mécaniques du véhicule	6
2.2 Unité de commande informatique embarquée	6
2.3 Précision de l'odométrie	9
2.4 Paramètres du modèle mesurés expérimentalement	13
CHAPITRE 3	18
CHAPITRE 4	32
CHAPITRE 5	41
5.1 Structure dans le signal image	47
CHAPITRE 6	53
6.1 Précision du profil	57
CHAPITRE 7	63

LISTE DES SYMBOLES

$A, B, C:$	Matrice d'état du système en translation,
$f:$	Coefficient de frottement visqueux,
$f_u, f_v:$	Distance focale,
$F_1, F_2:$	Forces de propulsion équivalentes,
$d:$	Déplacement d'un point d'une image,
$G:$	Matrice de gain de l'estimateur d'état,
$K_1, K_2:$	Gain du contrôleur tout-ou-rien,
$K_{a_1}, K_{a_2}:$	Gains des amplificateurs,
$K_{eq}:$	Gain de la fonction de transfert entre ϕ et ϕ_{ref} ,
$K_{rot_i}:$	Gains du contrôleur à retour d'état,
$K_{dir}:$	Gain au niveau de la direction,
$K_{tr}:$	Gain au niveau de la translation,
$K_{tri}:$	Gains du contrôleur du système en translation,
$K_{\phi_1}, K_{\phi_2}:$	Constantes de couple des moteurs,
$l:$	Longueur de l'entreaxe,
$l_i:$	Position intermédiaire le long du chemin,
$l_T:$	Longueur totale du chemin,
$L:$	Longueur du véhicule,
$M:$	Masse du véhicule,
$M_g:$	Vecteur représentant le modèle global,
$n_1, n_2:$	Rapports d'engrenage,
$p_i:$	Pôles du contrôleur à retour d'état,
$P:$	Position du véhicule le long du chemin,
$P_i:$	Point du profil global,

P_{ij} :	Point de la carte globale.
P_r :	Vecteur représentant le profil local,
P_{ref} :	Consigne de position du véhicule le long du chemin.
r, c :	Coordonné d'un point dans l'image.
r_{ij} :	Élément de la matrice de rotation.
r_o, c_o :	Centre de l'image,
R :	Matrice de rotation,
R_{a1}, R_{a2} :	Rayons des roues,
R_g :	Rayon de giration,
t_i :	Élément du vecteur translation,
T :	Vecteur translation,
u, v :	Coordonné du point dans le référentiel O' ,
u_{ij} :	Coefficient de mesure de l'appartenance,
U :	Matrice de partition,
U_{dir} :	Commande du moteur de direction,
U_{eq} :	Commande appliquée au contrôleur tout-ou-rien,
U_{tr1}, U_{tr2}, U_{tr} :	Commandes appliquées aux moteurs de propulsions.
V :	Vitesse de translation du véhicule au niveau de l'essieu arrière,
V_{ref} :	Consigne de vitesse de translation,
x :	Position en x du véhicule selon l'axe X,
x_c, y_c, z_c :	Coordonnées d'un point dans le référentiel de la camera,
x_k, y_k, z_k :	Coordonnées d'un point à l'instant k ,
x_m, y_m, z_m :	Coordonnées d'un point dans le référentiel monde.
x_r, y_r, z_r :	Coordonnées d'un point dans le référentiel du robot,

$^c x, ^c y$:	Position du véhicule dans le référentiel du chemin,
$^c X$:	Position en x du véhicule dans le référentiel du chemin,
X_i :	Vecteur de caractéristiques,
X_k :	Coordonnées d'un point à l'instant k ,
y :	Position en y du véhicule selon l'axe Y,
δ_a :	Déplacement élémentaire,
δ_d, δ_g :	Déplacement élémentaire de la roue droite et gauche,
δ_θ :	Rotation élémentaire,
ϵ_y :	Erreur latérale,
ϵ_θ :	Erreur d'orientation,
γ_1, γ_2 :	Couples appliqués aux roues,
ξ :	Facteur d'ammortissement,
ω_n :	Fréquence naturelle,
ϕ :	Orientation des roues directrices par rapport à l'axe du véhicule,
ϕ_{ref} :	Consigne d'orientation des roues directrices,
ϕ_o, Γ_o :	Matrice d'état discrétisé se l'observateur,
ρ :	Rayon de courbure du chemin,
τ_{dir} :	Constante de temps au niveau de la direction,
τ_{eq} :	Constante de temps de la fonction de transfert entre ϕ et ϕ_{ref} ,
θ :	Orientation du véhicule par rapport à l'axe X,
$^c \theta$:	Orientation du véhicule dans le référentiel du chemin,
θ_{ref} :	Consigne d'orientation du véhicule,

CHAPITRE 1

Introduction

Stationner un véhicule est une tâche complexe. Même avec beaucoup de pratique, plusieurs personnes ne peuvent accomplir facilement cette manœuvre. Pourtant, c'est une tâche réalisable et qui peut être automatisée. Un système de stationnement autonome doit avoir la capacité:

- d'identifier l'espace de stationnement,
- de planifier un chemin pour effectuer le stationnement en évitant les obstacles,
- de suivre le chemin planifié.

Chacun de ces problèmes a été étudié de façon plus ou moins approfondie dans la littérature portant sur la robotique mobile et la vision par ordinateur. Cependant, il existe peu d'exemples où des algorithmes de stationnement automatisé ont été implantés et testés expérimentalement. Par exemple, les réseaux neuronaux ont été utilisés pour tenté de résoudre le problème du stationnement d'un camion remorque [24] et celui du stationnement d'une automobile [25]. L'utilisation de la logique floue pour résoudre le problème de stationnement d'une automobile a également été étudiée dans [26].

Dans ce travail, la conception et l'intégration d'un système autonome de stationnement sont présentées. Ce système est basé sur des algorithmes qui résolvent les trois problèmes mentionnés précédemment, soit l'identification de la place de stationnement, la planification et le suivi de chemin. Les différents modules du

système sont illustrés sur la figure 1.1. Le robot parcourt la rangée de véhicules stationnés afin de construire une carte globale du profil de stationnement (figure 1.2). A partir de cette carte la manœuvre de stationnement est planifiée par un planificateur de chemin non-holonyme. Le chemin ainsi généré est alors suivi par le robot. Lors de l'exécution, le robot est relocalisé dans son environnement en faisant la mise en correspondance d'un profil local avec la carte globale. Le système proposé suppose qu'il y a une rangée d'automobile stationnées le long de la rue et que l'environnement est statique.

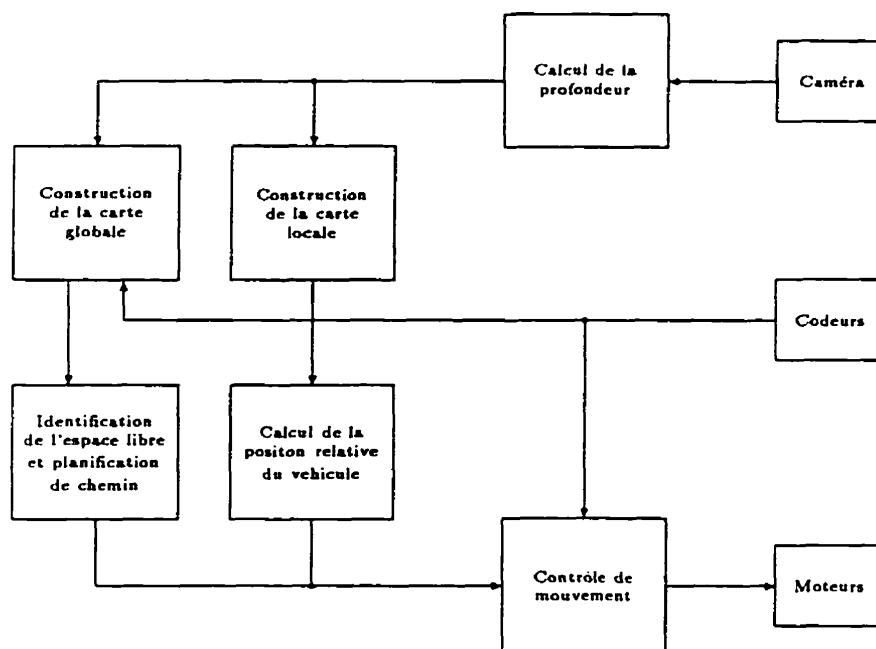


Figure 1.1 Module informatique du système de stationnement

Les chapitres qui suivent décrivent en détail chacune des étapes. Le chapitre 2 présente d'abord une description détaillée du véhicule. Les caractéristiques et les performances globales sont discutées en les comparant aux paramètres de conception. Un modèle dynamique simplifié du véhicule est ensuite présenté.

Dans le chapitre 3, les fonctions de base de contrôle de mouvement sont

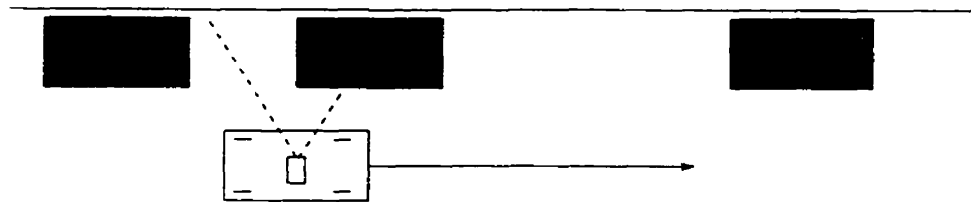


Figure 1.2 Scène

discutées ainsi que le problème du suivi de chemin.

Dans le chapitre 4, deux méthodes de planifications de chemin avec obstacles sont décrites et implantées. Tout d'abord nous présentons la méthode proposée par Barraquant et Latombe [8], puis la méthode de Laumond et Taix [10].

Le chapitre 5 présente une description du système de vision utilisé. Ce système de vision nous permet à partir d'une séquence d'image, d'obtenir de l'information télémétrique.

Le chapitre 6 présente une stratégie pour la construction de la carte globale représentant la scène, l'identification de l'espace de stationnement ainsi que la relocalisation du véhicule.

Puis finalement le dernier chapitre présente les conclusions de cette étude et discute des perspectives de recherche.

CHAPITRE 2

Caractérisation et modélisation

2.1 Introduction

Le système que nous allons développer dans les chapitres suivants a pour fonction d'automatiser le stationnement d'un véhicule automobile. Pour mettre au point les stratégies de contrôle et évaluer les performances par voie de simulation, un véhicule modèle réduit doit être développé.

Dans ce chapitre nous allons d'abord présenter le véhicule modèle réduit développé, son modèle dynamique ainsi que ses équations odométriques. Par la suite, le modèle est linéarisé et découplé pour simplifier la conception des contrôleurs de mouvement.

2.2 Conception du véhicule

Au cours des dernières années, de nombreux travaux de recherche ont porté sur l'étude et la mise au point des véhicules autonomes. Ces véhicules doivent avoir la capacité de percevoir leur environnement immédiat, de planifier leurs tâches et le chemin à suivre et d'exécuter ces mouvements en présence d'obstacles éventuels. Ces études impliquent de nombreux essais expérimentaux pour valider les différentes approches proposées et évaluer leurs performances. Les robots mobiles disponibles sur le marché, ou développés par les différents groupes de recherche pour réaliser ces tests, sont pour la plupart des plate-formes mobiles [1] ou des

véhicules à trois roues [2]. Ces robots mobiles sont en général assez encombrants, se déplacent à des vitesses inférieures à 1 m/s et ne sont pas représentatifs au niveau cinématique et dynamique des véhicules automobiles ou industriels (véhicule avec remorque, véhicule articulé).

Dans ce contexte, un véhicule automobile de taille réduite a été développé. Ce véhicule automobile baptisé VIGAR (Vision Guided Autonomous Robot) est montré à la figure 2.1. Ces spécifications sont les suivantes:

- Une longueur de 60 cm;
- Un poids de 32.8 kg;
- Une vitesse maximale de 3 m/s;
- Une accélération maximale de 3 m/sec².

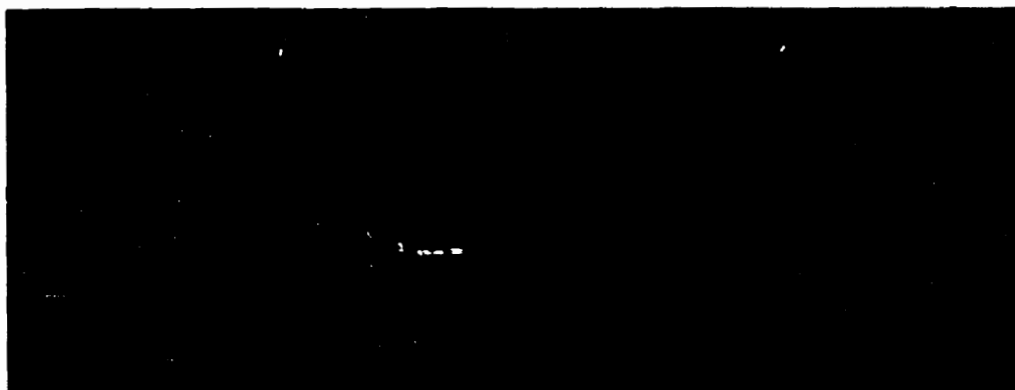


Figure 2.1 VIGAR

Le système d'entraînement retenu comprend deux moteurs CC pouvant fournir un couple de 2,5 N.m à 573 rpm, pour des roues possédant un rayon de 0,076 m.

Le tableau 2.1 donne la liste des composants électro-mécaniques du véhicule et le tableau 2.2 donne les éléments du système de commande informatique.

Tableau 2.1 Composants électro-mécaniques du véhicule

Moteurs de propulsions de direction de la tourelle	2 moteurs cc Pittman GM14605, 24Vcc, réducteur 4:1 1 moteur cc Pittman GM8712G276, 24Vcc, réducteur 96:1 2 moteurs pas-à-pas SLO-SYN M061-LS08
Amplificateurs	1 Copley controls PWM servo amplifier model 303 2 Copley controls PWM servo amplifiers model 306 SLO-SYN Micro series motion control: 2 translator modules 430-TH
Capteurs	3 Codeurs optiques HP HEDS-6000, 1000 et 500 imp/tour 3 HCTL-2000 Quadrature decoder / counter interface
Alimentation	2 Batteries GS Portalac PE6.5-12R, 12V, 6.5 Ah 1 Batterie YUASA NP12-12, 12V, 12 Ah convertisseur DC-DC, WR12T05-12/55K-3

Tableau 2.2 Unité de commande informatique embarquée

Ziatech ZT-8902 Ziatech ZT-8951 Ziatech ZT85CT61 Sensoray 7405 Ziatech ZT-200	carte CPU 486SX, 20MHz, 4Moctets de mémoires vives Unité de disquette 3,5 po et contrôleur 96 entrée-sorties numériques 8 convertisseurs N/A, boîtier STDbus 32 bits,
---	---

L'unité de commande informatique embarquée est composée d'un micro-ordinateur 486 de Ziatech (STD32). Le véhicule est relié par deux liens sériels (RS-232C) à un ordinateur-hôte IBM-PS2 et à une station de travail de type SUN SPARC. La figure 2.2 montre l'architecture globale du système de commande informatique. Comme il est indiqué sur la figure, le traitement des données

provenant de la caméra est réalisé sur une station de travail SUN.

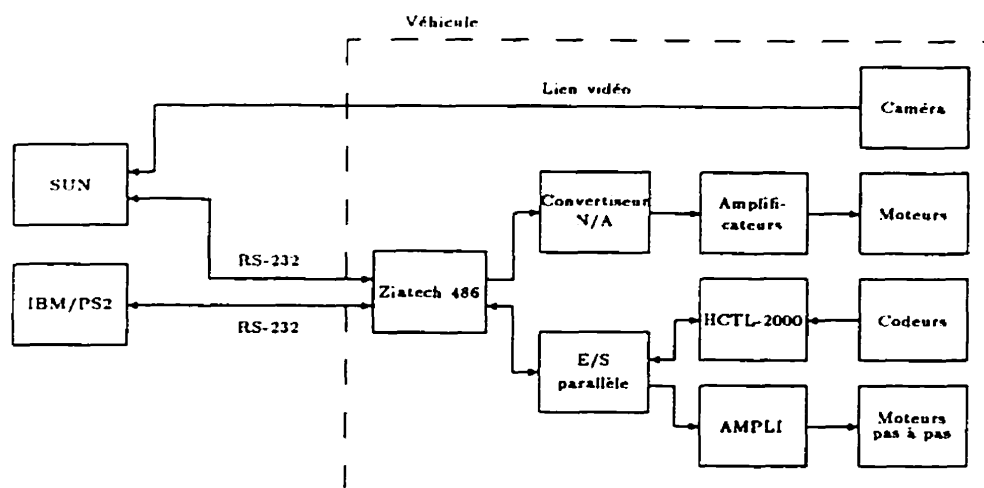


Figure 2.2 Architecture matériel du système informatique

Au niveau de la commande, seules les fonctions de contrôle de mouvement sont implantées au niveau du micro-ordinateur embarqué 486. Le traitement des données télémétriques comme le calcul de la profondeur, la construction de cartes locale et globale, ainsi que les fonctions de planification de chemin sont réalisés sur la station de travail SUN.

2.3 Odométrie

La localisation par odométrie est basée sur l'intégration des informations de mouvements internes aux robots. Les capteurs de mouvement du véhicule automobile sont des codeurs optiques montés sur les roues motrices et sur la direction. Pendant le mouvement du robot la position est mise à jour par intégration des déplacements élémentaires des roues. De ce fait l'odométrie est faussée s'il y a glissement ou patinage. Nous allons présenter les équations donnant la position et l'orientation courantes du robot par intégration des mouvements élémentaires.

Pour le cas du véhicule automobile les déplacements élémentaires sont rectilignes ou circulaires. Un déplacement élémentaire est représenté à la figure 2.3.

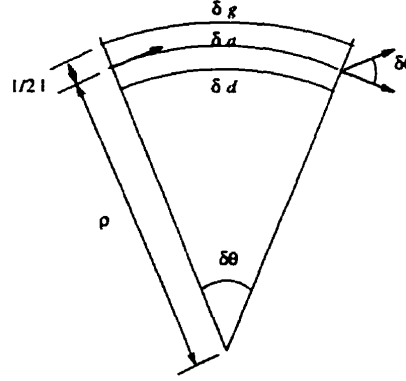


Figure 2.3 Déplacement élémentaire

La rotation élémentaire du robot est:

$$\delta\theta = \frac{\delta d - \delta g}{l} \quad (2.1)$$

où: δg et δd : déplacement élémentaire de chaque roue;

l : longueur de l'entreaxe.

Le déplacement élémentaire est:

$$\delta a = \frac{\delta d + \delta g}{2} \quad (2.2)$$

Soit x_n , y_n et θ_n la configuration courante du véhicule à la période n . À partir des cinq valeurs x_n , y_n , θ_n , δa_n et $\delta\theta_n$ on peut calculer la configuration à la prochaine période d'échantillonnage $(n + 1)$.

$$\begin{aligned} x_{n+1} &= x_n + \delta a_n \cos(\theta_n + \frac{\delta\theta_n}{2}) \\ y_{n+1} &= y_n + \delta a_n \sin(\theta_n + \frac{\delta\theta_n}{2}) \\ \theta_{n+1} &= \theta_n + \delta\theta_n \end{aligned} \quad (2.3)$$

Nous avons supposé, pour faire l'intégration, la trajectoire élémentaire rectiligne ou circulaire. Ceci est vrai pour un temps dt infinitésimal. De plus pour la trajectoire circulaire on suppose que l'arc de cercle est égal à la corde. Comme nous discrétisons en remplaçant dt par la période d'échantillonnage T , une erreur est engendrée en supposant une courbure constante durant cette période d'échantillonnage.

Malgré les hypothèses utilisées la précision des équations est satisfaisante pour une zone de travail limitée. Il est à remarquer que seule l'information provenant des roues motrices a été utilisée pour développer les équations odométriques, l'information provenant de la direction n'étant pas utilisée. Ce choix est motivé par le fait qu'il est difficile de mesurer précisément l'angle des roues directrices. En effet, il est difficile de faire une initialisation précise de la position de la direction et il existe un jeu non négligable au niveau du mécanisme de la direction. L'erreur de l'odométrie pour une trajectoire rectiligne et une trajectoire circulaire est présentée au tableau 2.3, les valeurs indiqués sont des moyennes calculées sur 5 essais. Ces tests ont été effectués en poussant le véhicule à la main et pour une période d'échantillonnage $T = 10$ ms.

Tableau 2.3 Précision de l'odométrie

Trajectoire	Erreur		
	x	y	θ
Rectiligne (2m)	0.001m	0.005m	0.014°
Circulaire (Demi cercle de rayon 0.7m)	0.05m	0.03m	0.06°

2.4 Modèle dynamique du véhicule

Le modèle utilisé pour analyser le comportement du véhicule s'inspire du modèle simplifié développé dans [3] pour un véhicule à deux roues motrices. Pour ce type de véhicule, on néglige la dynamique reliée au roulis et au tangage ainsi que les effets associés à la déformation et au glissement des pneus. Ce modèle est alors valable pour des vitesses de déplacement et des accélérations latérales faibles. En tenant compte de ces hypothèses simplificatrices, le modèle est développé en considérant la cinétique du véhicule auquel on ajoute le modèle des actionneurs. A partir du schéma de la figure 2.4, on peut écrire les équations décrivant la cinétique du véhicule [6]:

$$\begin{aligned}\dot{x} &= V \cos \theta \\ \dot{y} &= V \sin \theta \\ \dot{\theta} &= V \frac{\tan \phi}{L}\end{aligned}\tag{2.4}$$

- où x : position en x du véhicule selon l'axe X ,
 y : position en y du véhicule selon l'axe Y ,
 θ : orientation du véhicule par rapport à l'axe X ,
 ϕ : orientation des roues directrices par rapport à l'axe du véhicule,
 V : vitesse de translation du véhicule au niveau de l'essieu arrière.

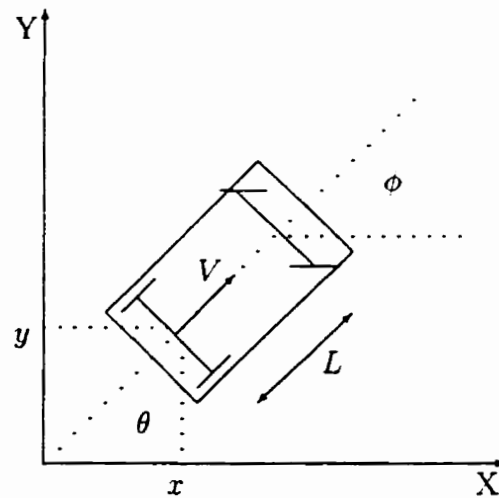


Figure 2.4 Représentation du véhicule dans le plan XY

La dynamique du véhicule en translation en fonction des tensions appliquées aux bornes des servomoteurs est obtenue en combinant l'équation de Newton

$$F_1 + F_2 = M\dot{V} + fV \quad (2.5)$$

et celle du groupe amplificateur-moteur (notons que l'amplificateur utilisé est de type tension-courant):

$$\Gamma_i = K_{\phi_i} K_{a_i} U_{tr_i} \quad (2.6)$$

De plus

$$F_i = \frac{\Gamma_i}{n_i R_{a_i}} \quad (2.7)$$

avec: U_{tr1}, U_{tr2} : Commandes appliquées aux moteurs;
 K_{a1}, K_{a2} : Gains des amplificateurs (2 amp/volt);
 $K_{\phi1}, K_{\phi2}$: Constantes de couple des moteurs (0.0612 N-m/amp);
 Γ_1, Γ_2 : Couples appliqués aux roues;
 F_1, F_2 : Forces de propulsion équivalente;
 R_{a1}, R_{a2} : Rayons des roues (0.07625 m);
 n_1, n_2 : Rapports d'engrenage (0.25);
 M : Masse de véhicule (32.8 kg);
 f : Coefficient de frottement visqueux.

En regroupant les termes et en négligeant le frottement visqueux, la dynamique du véhicule en translation est représentée par un intégrateur avec un gain K_{tr} soit:

$$\dot{V} = K_{tr} U_{tr} \quad (2.8)$$

avec

$$K_{tr} = \frac{2K_a K_\phi}{n R_a M} = 0.346 \text{ m/s}^2 \text{ volts.} \quad (2.9)$$

Finalement un modèle du premier ordre est utilisé pour représenter la dynamique de l'actionneur au niveau de la direction du système (les moments d'inertie des roues et du véhicule sont négligés). On a donc

$$\ddot{\phi} = \frac{K_{dir}}{\tau_{dir}} U_{dir} - \frac{\dot{\phi}}{\tau_{dir}} \quad (2.10)$$

avec: U_{tr} : commande des moteurs de propulsion,
 U_{dir} : commande du moteur de direction,
 K_{tr} : gain au niveau de la translation,
 K_{dir} : gain au niveau de la direction,
 τ_{dir} : constante de temps au niveau de la direction.

Le modèle complet est donc représenté à la figure 2.5. Les gains K_{dir} , K_{tr} et la constante de temps τ_{dir} ont été identifiés à partir d'essais expérimentaux (réponse à l'échelon) et leurs valeurs sont données dans le tableau 2.4. Il est à remarquer que le gain en translation mesuré est très proche du gain théorique (équation 2.9) malgré le fait que l'on a négligé le frottement visqueux.

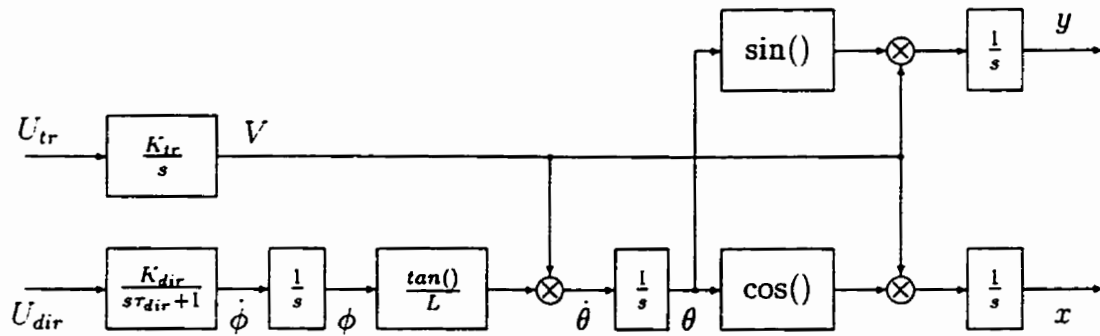


Figure 2.5 Modèle dynamique

Tableau 2.4 Paramètres du modèle mesurés expérimentalement

K_{dir}	8218 deg/s ² volt
τ_{dir}	3 ms
K_{tr}	0.350 m/s ² volt

2.5 Problème de suivi de chemin

Il est essentiel d'élaborer un contrôleur de mouvements pour le véhicule de façon à pouvoir suivre des chemins pré-définis dans le plan XY . Comme il est discuté par DeSantis dans [4] [5], le problème de suivi de chemin pour un robot mobile, peut être traité suivant deux approches: le suivi de trajectoire désirée au niveau des états du système ou le suivi de chemin géométrique. Dans le cas du suivi de trajectoire, le contrôleur doit assurer la convergence des états du véhicule vers les états désirés qui sont fonction du temps. Cette approche conduit pour un véhicule représenté par un modèle non-linéaire, à des contrôleurs complexes et qui utilisent des stratégies qui ne sont pas conformes aux pratiques utilisées par un conducteur. En effet l'objectif d'un conducteur est de maintenir le véhicule le plus près possible du milieu de la route et ceci indépendamment du temps de parcours. Dans ce cas l'erreur que le conducteur corrige dépend de la localisation du véhicule par rapport au chemin à suivre. Il ne fait pas varier la vitesse de son véhicule de façon à ce qu'au temps t_1 elle soit à la position p_1 sur le chemin. Cette stratégie de commande est basée sur la notion de suivi géométrique de chemin.

Dans ce contexte l'erreur à corriger est définie à la figure 2.6. Cette erreur est constituée d'une erreur latérale (ϵ_y), d'une erreur d'orientation (ϵ_θ) et d'une erreur de direction (ϵ_ϕ).

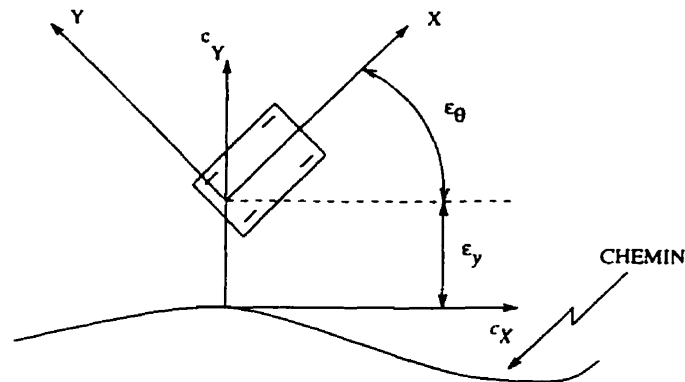


Figure 2.6 Référentiel du chemin

A partir de la figure 2.6 on peut donc écrire que:

$$\begin{aligned}
 \epsilon_\theta &= \theta_{ref} - \theta \\
 \dot{\epsilon}_\theta &= \dot{\theta}_{ref} - \frac{\tan \phi}{L} V \\
 \dot{\epsilon}_y &= V \sin \epsilon_\theta \\
 {}^c\dot{X} &= V \cos \epsilon_\theta
 \end{aligned} \tag{2.11}$$

L'erreur de direction ϵ_ϕ est la différence entre l'angle d'orientation des roues du véhicule (ϕ) et entre l'angle correspondant au rayon de courbure du chemin (ϕ_{ref}), soit:

$$\epsilon_\phi = \phi - \phi_{ref} \tag{2.12}$$

En combinant les équations 2.8, 2.10 et 2.11 on obtient alors le modèle de l'erreur de suivi de chemin suivant:

$$\begin{aligned}
{}^c\dot{X} &= V \cos \epsilon_\theta \\
\dot{\epsilon}_y &= V \sin \epsilon_\theta \\
\dot{\epsilon}_\theta &= -V \frac{\tan \phi}{L} + \dot{\theta}_{ref} \\
\dot{\phi} &= \dot{\phi} \\
\ddot{\phi} &= -\frac{\dot{\phi}}{\tau_{dir}} + \frac{K_{dir}}{\tau_{dir}} U_{dir} \\
V &= K_{tr} U_{tr}
\end{aligned} \tag{2.13}$$

En posant les hypothèses suivantes: l'erreur d'orientation ϵ_θ est petite, la vitesse du véhicule est approximativement égale à la vitesse de référence ($V \cong V_{ref}$) et l'angle des roues est faibles ($\phi < 34^\circ$), on obtient les modèles linéarisés découplés pour le véhicule:

Modèle du véhicule en translation:

$$\begin{bmatrix} {}^c\dot{X} \\ \dot{V} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} {}^cX \\ V \end{bmatrix} + \begin{bmatrix} 0 \\ K_{tr} \end{bmatrix} U_{tr}. \tag{2.14}$$

Modèle du véhicule en rotation:

$$\begin{bmatrix} \dot{\epsilon}_y \\ \dot{\epsilon}_\theta \\ \dot{\phi} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & V_{ref} & 0 & 0 \\ 0 & 0 & \frac{-V_{ref}}{L} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & \frac{-1}{\tau_{dir}} \end{bmatrix} \begin{bmatrix} \epsilon_y \\ \epsilon_\theta \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{K_{dir}}{\tau_{dir}} \end{bmatrix} U_{dir} + \begin{bmatrix} 0 \\ \dot{\theta}_{ref} \\ 0 \\ 0 \end{bmatrix} \tag{2.15}$$

Ces modèles linéaires et découplés permettent de concevoir les contrôleurs de mouvement.

2.6 Conclusion

Dans ce chapitre nous avons d'abord présenté le robot mobile qui a été construit pour réaliser l'expérimentation. Ce robot de petite taille est représentatif au niveau cinématique et dynamique du comportement des véhicules automobiles à basse vitesse. L'utilisation d'une architecture incluant un micro-ordinateur embarqué pour l'implantation des fonctions de contrôle fait de ce robot un banc d'essai idéal pour l'expérimentation. Nous avons développé les équations odométriques et le modèle dynamique de ce robot. Des essais expérimentaux ont démontré la validité des équations odométriques. Seule la validité du modèle dynamique en translation a été testé expérimentalement. Comme ce modèle dynamique néglige la dynamique de rotation, il est donc valide que pour des accélérations et des vitesses latérales faibles. Afin de concevoir, au chapitre suivant, des contrôleurs simples pour le suivi de chemin géométrique, le modèle dynamique a finalement été linéarisé et découplé. Le modèle pour la conception des contrôleurs est constitué de deux sous-systèmes: un système en translation et un système en rotation.

CHAPITRE 3

Contrôle de mouvement

3.1 Introduction

La stratégie de contrôle de mouvement doit permettre au véhicule de suivre un chemin à une vitesse de consigne V_{ref} . Comme notre système est découplé, une commande décentralisée, c'est-à-dire un contrôleur pour chaque sous-système, est appropriée. Dans ce chapitre un estimateur d'état qui permet de filtrer les données odométriques est d'abord développé. Par la suite les deux contrôleurs de mouvements sont présentés: le contrôleur du système en translation et le contrôleur de la direction. Finalement, les résultats de simulation et expérimentaux sont présentés pour la correction d'erreur latérale, la correction d'erreur d'orientation et dans le cas général d'un suivi de chemin quelconque.

3.2 Filtrage des données odométriques

La position angulaire des moteurs du robot mobile est obtenue à partir des codeurs optiques. On utilise la méthode des différences finies à des intervalles de temps réguliers pour obtenir la vitesse angulaire du moteur. On constate qu'avec cette méthode, à mesure que la période d'échantillonnage diminue, la mesure de vitesse se dégrade. La vitesse devient donc entachée d'une quantité importante de bruit.

Pour le système en translation du robot mobile, nous devons concevoir un contrôleur de vitesse. Le bruit sur la mesure de vitesse va donc limiter les

gains du contrôleur. Pour diminuer le bruit, un estimateur d'état est utilisé [22].

L'estimateur d'état (figure 3.1) est donné par l'équation suivante:

$$\dot{\tilde{x}}(t) = A\tilde{x}(t) + Bu(t) + G(y(t) - \tilde{y}(t)) \quad (3.1)$$

$$\dot{\tilde{x}}(t) = (A - GC)\tilde{x}(t) + Bu(t) + Gy(t). \quad (3.2)$$

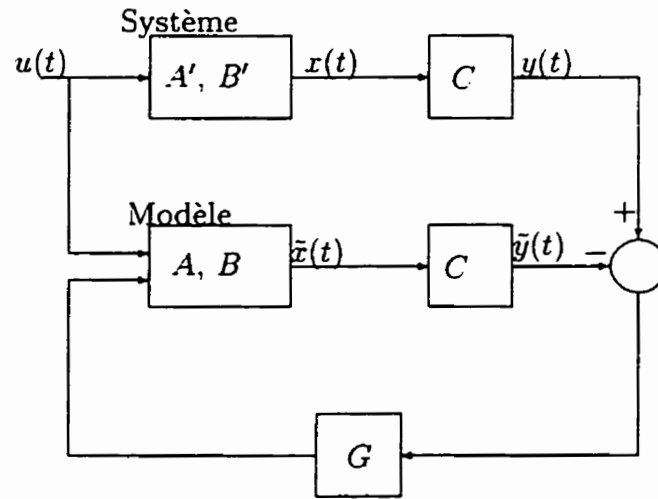


Figure 3.1 Estimateur d'état

Les matrices A, B, C sont données par le modèle d'état du système en translation:

$$\begin{bmatrix} {}^c\dot{X} \\ \dot{V} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} {}^cX \\ V \end{bmatrix} + \begin{bmatrix} 0 \\ K_{tr} \end{bmatrix} U_{tr} \quad (3.3)$$

$${}^cX = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} {}^cX \\ V \end{bmatrix}. \quad (3.4)$$

L'équation caractéristique désirée de l'erreur d'estimation s'écrit:

$$\alpha_e(s) = s^2 + \frac{2\xi}{\omega_m} s + \frac{1}{\omega_m^2}, \quad (3.5)$$

où $\xi = 1$. La matrice de gain G est donnée par [23]:

$$G = \alpha_e(A) \begin{bmatrix} C \\ CA \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (3.6)$$

où

$$\alpha_e(A) = \begin{bmatrix} \frac{1}{\omega_m^2} & \frac{2}{\omega_m} \\ 0 & \frac{1}{\omega_m^2} \end{bmatrix} \quad (3.7)$$

avec

$$\begin{bmatrix} C \\ CA \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \quad (3.8)$$

Ainsi on obtient

$$G = \begin{bmatrix} \frac{2}{\omega_m} \\ \frac{1}{\omega_m^2} \end{bmatrix} \quad (3.9)$$

Pour implanter l'observateur on doit discrétiser l'équation 3.2 [21] et on obtient:

$$\begin{aligned} \bar{x}_{i+1} &= \Phi_o \hat{x}_i + \Gamma_o u_i \\ \hat{x}_{i+1} &= \bar{x}_{i+1} + G(y_i - \hat{y}_i) \\ \hat{y}_{i+1} &= C \hat{x}_{i+1} \end{aligned} \quad (3.10)$$

où Φ_o et Γ_o sont les matrices d'état discrétisées de l'observateur.

La figure 3.2 représente la mesure de vitesse avant le filtrage (graphique du haut) et la vitesse après filtrage (graphique du bas) pour un $\omega_m = 0.02$. Les essais expérimentaux du contrôleur de vitesse ont permis de choisir cette valeur

de ω_m qui permet d'obtenir une mesure de qualité suffisante, c'est-à-dire que le bruit est atténué sans que les variations réelles de la vitesse soient détériorées.

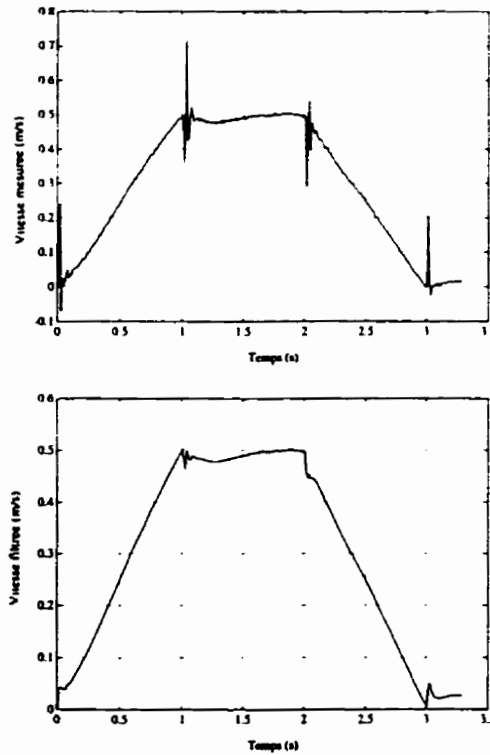


Figure 3.2 Vitesse avant filtrage et vitesse filtrée

3.3 Contrôleur de position et de vitesse pour le système en translation

Au niveau des déplacements en translation du véhicule, le contrôleur doit réaliser deux fonctions. La première est de régulariser la vitesse du véhicule le long du chemin (mode croisière) et la deuxième est de positionner le véhicule à un endroit précis dans l'espace de travail (mode positionnement).

Il s'agit donc d'un contrôleur de position et de vitesse pour le système en

translation. Il est possible d'utiliser un contrôleur de type PI anticipatif pour la vitesse et PI pour la position, selon la loi de commande suivante (figure 3.3):

$$U_{tr} = K_{tr1}(V_{ref} - V) + K_{tr2} \int (V_{ref} - V) dt + \frac{1}{K_{tr}} \frac{d(V_{ref})}{dt} + (K_{tr3}(P_{ref} - P) + K_{tr4} \int (P_{ref} - P) dt) * S1 \quad (3.11)$$

avec: V_{ref} étant la vitesse désirée.

P_{ref} étant la position désirée.

$S1$ étant 0 ou 1.

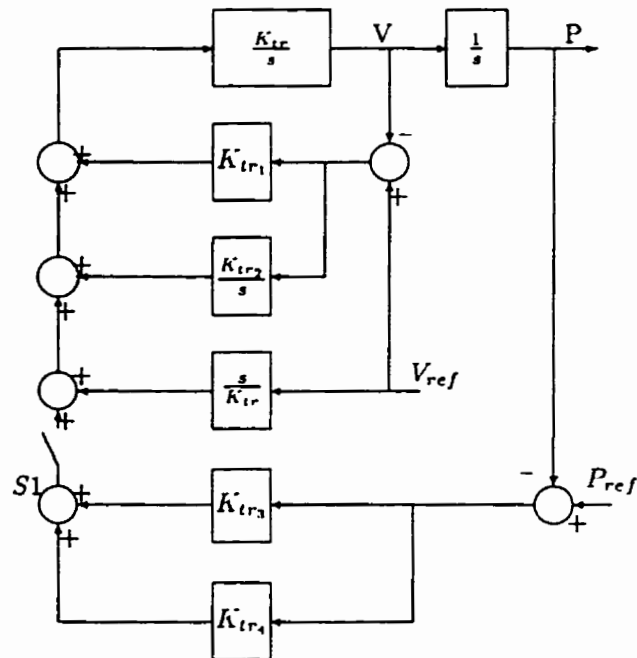


Figure 3.3 Contrôleur du système en translation

On utilise ce contrôleur avec une consigne de position et de vitesse. La consigne de vitesse doit être la dérivée de la consigne de position. La consigne de vitesse (figure 3.4) utilisée est un trapèze. En régime de croisière on utilise seulement le régulateur de vitesse.

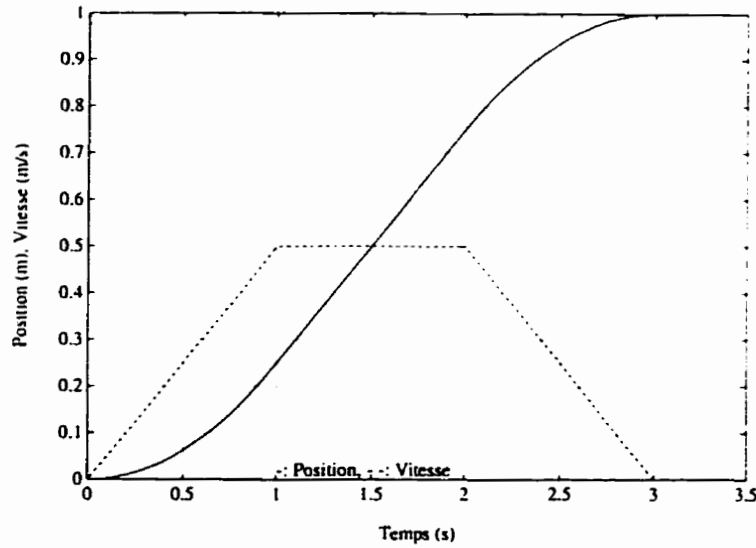


Figure 3.4 Consigne de vitesse et de position

Pour éviter le glissement des roues motrices, une consigne différente est utilisée pour la roue droite et la roue gauche lorsque l'angle des roues directrices n'est pas nulle. On calcule les consignes pour les roues motrices en utilisant le rapport entre la longueur de l'entreaxe l et le rayon de giration R_g du véhicule.

$$\begin{aligned} V_{gref} &= (1 + \frac{l}{2R_g})V_{ref} & P_{gref} &= (1 + \frac{l}{2R_g})P_{ref} \\ V_{dref} &= (1 - \frac{l}{2R_g})V_{ref} & P_{dref} &= (1 - \frac{l}{2R_g})P_{ref} \end{aligned} \quad (3.12)$$

Les valeurs des gains sont choisies expérimentalement par essais successifs. Les valeurs des gains $K_{tr1} = 20$, $K_{tr2} = 5$, $K_{tr3} = 50$ et $K_{tr4} = 10$ donnent des performances satisfaisantes. Les résultats obtenus pour un déplacement de 1 m et de 0.1 m sont présentés aux figure 3.5 et 3.6. Les résultats nous montrent l'efficacité des contrôleurs. Dans les deux cas l'erreur en position est de l'ordre de 0.02 m. Cette erreur est principalement causée par la zone morte du moteur.

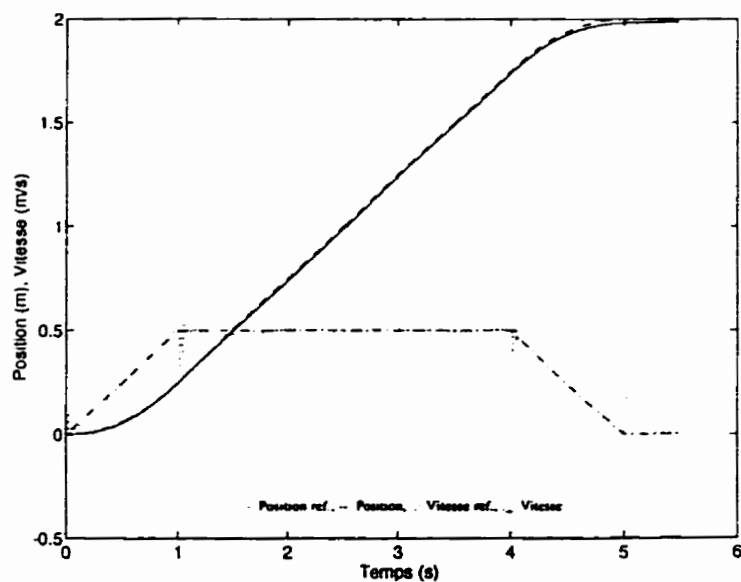


Figure 3.5 Déplacement de 2m

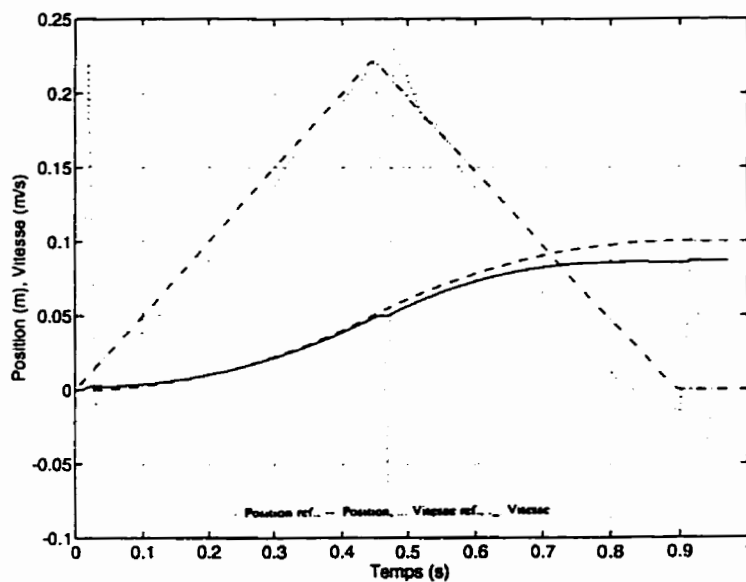


Figure 3.6 Déplacement de 0.1m

3.4 Correction d'erreur de suivi de chemin

Le contrôleur de la direction doit permettre de maintenir le véhicule le plus près possible du chemin; pour se faire il doit corriger l'erreur latérale, l'erreur

d'orientation et l'erreur de la direction définies par la figure 2.6. Dans le cas du suivi de chemin l'erreur latérale est définie par:

$$\epsilon_y = -\sin \epsilon_\theta (x - {}^c x) + \cos \epsilon_\theta (y - {}^c y), \quad (3.13)$$

l'erreur d'orientation par:

$$\epsilon_\theta = \theta - {}^c \theta \quad (3.14)$$

et l'erreur de la direction par:

$$\epsilon_\phi = \phi - \phi_{ref} \quad (3.15)$$

Ces erreurs sont corrigées en envoyant une commande U_{dir} au moteur de la direction.

Pour le contrôleur du système en rotation, ou plus simplement le contrôleur de position/orientation, deux contrôleurs imbriqués sont utilisés. Un premier contrôleur de type tout-ou-rien assure que l'orientation ϕ des roues directrices est égale à la valeur de consigne ϕ_{cons} (figure 3.7), selon la loi de commande suivante:

$$U_z = K_1(\phi_{cons} - \phi) + K_2\dot{\phi}$$

$$U_{dir} = \begin{cases} 0.5 & \text{si } U_z > 0.5 \\ -0.5 & \text{si } U_z < -0.5 \\ 0 & \text{si } -0.5 \leq U_z \leq 0.5 \end{cases} \quad (3.16)$$

Les gains de ce contrôleur sont choisis expérimentalement de façon à obtenir un temps de réponse de 10ms et un dépassement nul. Les valeurs de $K_1 = 1$ et $K_2 = 0.01$ donnent des performances satisfaisantes.

Par la suite, on suppose que la fonction de transfert entre ϕ et ϕ_{cons} est du premier ordre (avec $K_{eq} = 1$ et $\tau_{eq} = 10$ ms). La deuxième boucle de contrôle en rotation est alors une boucle de commande par retour d'état avec anticipation tel que représentée par la loi de commande suivante:

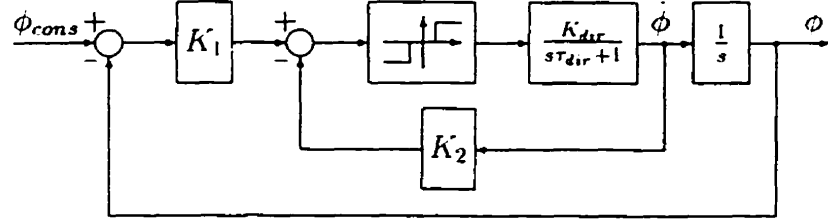


Figure 3.7 Contrôleur tout-ou-rien

$$\phi_{cons} = \frac{1}{K_{eq}} \phi_{ref} + K_{rot1} \epsilon_y + K_{rot2} \epsilon_\theta + K_{rot3} \epsilon_\phi \quad (3.17)$$

Les gains K_{roti} de cette boucle de commande sont ajustés en fixant les pôles décrivant la dynamique du système en boucle fermée représenté par les équations 2.14 et 3.17. Ainsi:

$$\begin{aligned} K_{rot1} &= -(p_1 p_2 p_3) \frac{\tau_{dir} L}{V_{ref}^2} \\ K_{rot2} &= (p_1 p_2 + p_1 p_3 + p_2 p_3) \frac{\tau_{dir} L}{V_{ref}} \\ K_{rot3} &= -(p_1 + p_2 + p_3) \tau_{dir} - 1 \end{aligned} \quad (3.18)$$

où: p_1 , p_2 et p_3 sont les pôles désirés. On remarque que le modèle linéaire du véhicule, et par conséquent les gains (K_{rot1} et K_{rot2}), dépendent de la vitesse de déplacement V_{ref} . Ainsi, ces gains seront mis à jour en fonction de la vitesse.

Les contrôleurs proposés ont été analysés en simulation et expérimentalement. Les résultats obtenus pour la correction d'une erreur latérale sont présentés à la figure 3.8. Les résultats pour la correction d'une erreur latérale et d'orientation à la figure 3.9. Ces résultats correspondent à des pôles désirés égaux à $p_1 = -0.8317$, $p_2 = -0.7035 + 0.9763 i$ et $p_3 = -0.7035 - 0.9763 i$.

Sur les figures les résultats en simulation (trait plein) et en expérimentation

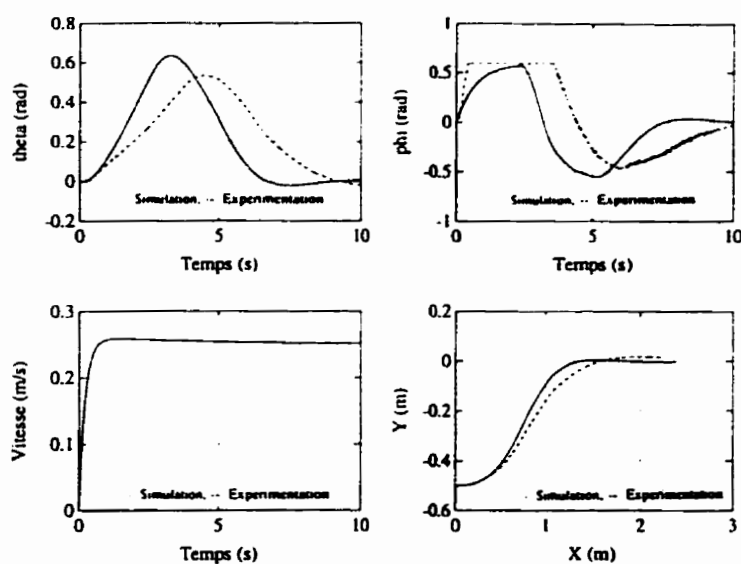


Figure 3.8 Correction d'une erreur latérale

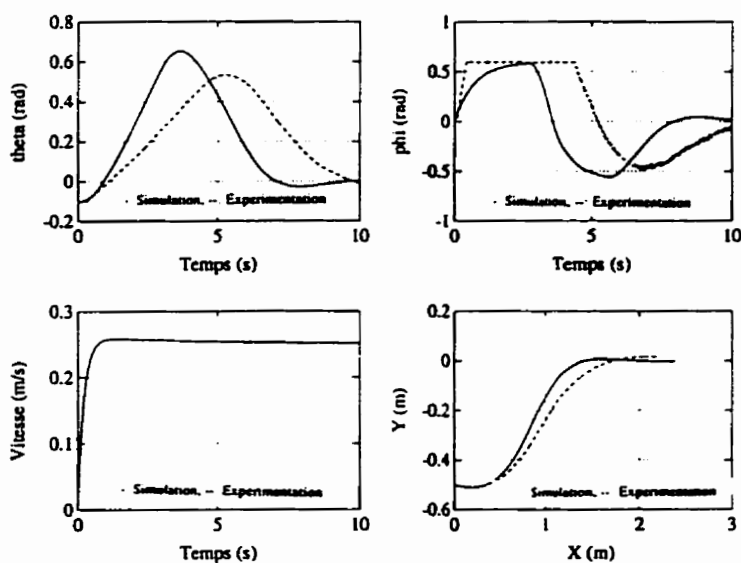


Figure 3.9 Correction d'une erreur latérale et d'orientation

(trait pointillé) sont présentés. On constate de façon générale que les contrôleurs corrigent relativement bien les erreurs latérale et d'orientation. Par contre on remarque une différence non négligable entre les résultats de simulation et les résultats d'expérimentation. Ces différences proviennent du fait que le modèle ne

décrit pas parfaitement le véhicule. En effet le modèle proposé ne tient pas compte de la dynamique de la rotation, de l'absence de points de contacts unique entre les roues et le sol. L'absence de suspension sur le robot ne nous permet pas de garantir que les quatre roues sont toujours en contact avec le sol. Un modèle plus élaboré permettrait d'obtenir des résultats similaires entre l'expérimentation et la simulation. Il est à noter que malgré tout ce modèle nous a permis de concevoir des contrôleurs satisfaisants.

3.5 Suivi de chemin

Maintenant que nous avons fait la conception des contrôleurs pour chacun des deux sous-systèmes, nous sommes prêts à tester la stratégie de commande élaborée pour réaliser un suivi de chemin. Les chemins que le robot doit suivre se composent d'une suite de configurations. Pour que le chemin soit réalisable, la suite de configurations qui le compose doit tenir compte des contraintes cinématiques du véhicule.

L'algorithme de suivi de chemin proposé fait suivre au véhicule l'axe des X des référentiels associés aux configurations qui décrivent le chemin. Ainsi l'erreur latérale et l'erreur d'orientation est tout simplement calculer en exprimant les vecteurs de configuration du robot dans le référentiel du point du chemin le plus près du robot. L'algorithme suivant est utilisé pour suivre un chemin composé d'une suite de configurations.

- Segmenter le chemin, en portion avance et recul.
- Pour chaque portion de chemin calculer les gains du contrôleur du système en rotation (équation 3.18).
- Suivi de chaque portion de chemin.

- Calculer la position du véhicule par odométrie (x, y, θ, ϕ)
- Trouver le point du chemin $(^c x_n, ^c y_n, ^c \theta_n)$ qui est le plus proche de la position du véhicule.
- Calculer le rayon de courbure du chemin pour déduire ϕ_{ref}

$$\rho = \frac{\sqrt{(^c x_{n+1} - ^c x_{n-1})^2 + (^c y_{n+1} - ^c y_{n-1})^2}}{2 \sin\left(\frac{(^c \theta_{n+1} - ^c \theta_{n-1})}{2}\right)} \quad (3.19)$$

$$\tan \phi = \frac{L}{\rho} \quad (3.20)$$

- Calculer l'erreur d'orientation

$$\epsilon_\theta = \theta - ^c \theta \quad (3.21)$$

- Calculer l'erreur latérale

$$\epsilon_y = -\sin \epsilon_\theta (x - ^c x) + \cos \epsilon_\theta (y - ^c y) \quad (3.22)$$

- Calculer la consigne U_{dir} à partir des équations 3.16 et 3.17
- Identifier si l'on se trouve en régime de croisière à partir du point du chemin qui est le plus proche de la position du véhicule.
- Calculer la consigne U_{tr} à partir de l'équation 3.11

Les figures 3.10 et 3.11 présentent les résultats d'expérimentation du suivi de différents chemins. L'erreur latérale et d'orientation finales pour l'essai de la figure 3.10 sont: 0.0079 m et 0.023° et pour l'essai de la figure 3.11, 0.005 m et 0.01°.

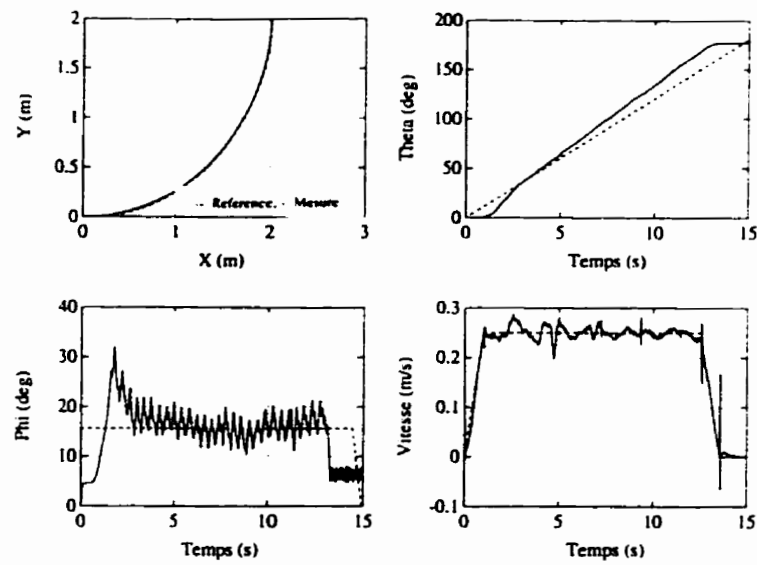


Figure 3.10 Suivi d'un arc de cercle

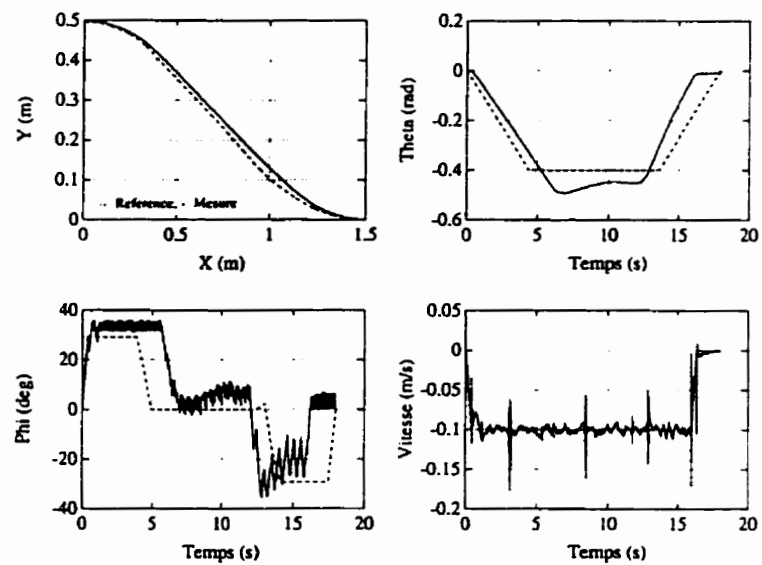


Figure 3.11 Suivi d'un créneau

3.6 Conclusion

Dans ce chapitre un estimateur d'état qui nous permet de filtrer les mesures provenant des codeurs optiques a été mis au point. Par la suite nous avons

développé une stratégie de commande qui nous permet de réaliser un suivi de chemin. Cette stratégie est basée sur une approche géométrique pour le suivi de chemin, c'est-à-dire que l'on désire corriger les erreurs latérale et d'orientation du véhicule par rapport au chemin désiré. Un contrôleur pour chaque sous-système a été proposé: l'un pour le système en translation et l'autre pour le système en rotation. Un contrôleur de position et de vitesse a été implanté pour le système en translation. Ce contrôleur permet de faire du suivi de chemin et du suivi de trajectoire. Comme dans le cas du stationnement il n'y a pas de régime de croisière nous utilisons ce contrôleur pour faire du suivi de trajectoire, mais il a été conçu de façon à permettre le suivi de chemin lorsque nécessaire. Des essais expérimentaux ont démontré que l'estimateur d'état permet d'obtenir une vitesse estimée de qualité suffisante pour être en mesure d'utiliser des gains assez élevés pour ce contrôleur et d'obtenir ainsi de bonnes performances. Un contrôleur du système en rotation a été mis au point pour corriger l'erreur entre le véhicule et le chemin à suivre. Malgré les différences entre les simulations et l'expérimentation, les contrôleurs ont démontré leur efficacité. Les erreurs obtenues en fin de parcours pour un chemin de type arc de cercle sont inférieures à 8 mm et 0.025° . Ceci nous permettra de suivre les chemins générés par les méthodes décrites au chapitre suivant.

CHAPITRE 4

Planification de chemin

4.1 Introduction

Ce chapitre présente les méthodes de planification de chemin étudiées pour stationner le véhicule. Le problème de planification de chemin consiste à déplacer un robot d'une configuration initiale à une configuration finale. La configuration d'une automobile (figure 2.4) est définie par la localisation du centre de l'essieu des roues arrières de l'automobile dans le plan XY , l'orientation θ de l'automobile par rapport à l'axe x et l'angle des roues directrices ϕ par rapport à l'axe de l'automobile.

Pour planifier un chemin entre deux configurations, il faut respecter les contraintes cinématiques du robot. Par exemple, une automobile possède des contraintes cinématiques puisqu'elle ne peut se déplacer latéralement ni tourner sur elle même. Mais nous savons qu'il est possible de la positionner à n'importe quelle position ou orientation dans le plan. Le stationnement d'une voiture est donc un exemple de problème de planification de chemin non-holonome. Pour le problème qui nous intéresse, il faut tenir compte à la fois des contraintes de mouvement du robot et des contraintes imposées par l'environnement.

Nous allons dans ce chapitre considérer deux méthodes de planification de chemin en présence d'obstacles. L'une proposée par Barraquand et Latombe [8] et l'autre proposée par Taix et Laumond [10].

4.2 Méthode de Barraquant et Latombe [9]

Cette méthode consiste tout d'abord à discrétiser l'espace de configuration en un ensemble de cellules élémentaires. La voiture se déplace entre les cellules en considérant que la vitesse V et l'orientation ϕ sont constantes. En obligeant V à prendre les valeurs $\{-V_{max}, V_{max}\}$ et ϕ à prendre les valeurs $\{-\phi_{max}, 0, \phi_{max}\}$, et en intégrant l'équation 2.4 sur un temps fixe on obtient six chemins constants. Étant donnée une configuration initiale appartenant à la liste de nœuds à explorer, l'algorithme génère, à l'aide des six chemins constants, les six configurations atteignables. Si la nouvelle configuration rencontre un obstacle ou se trouve à l'intérieur d'une cellule déjà explorée, alors cette configuration est rejetée. Autrement, elle est ajoutée à la liste de nœuds devant être explorés suivant l'ordre croissant d'une fonction de coût. De cette façon, les obstacles sont incorporés en même temps que le chemin réalisable est généré. Les obstacles sont utilisés pour réduire la taille de la structure arborescente.

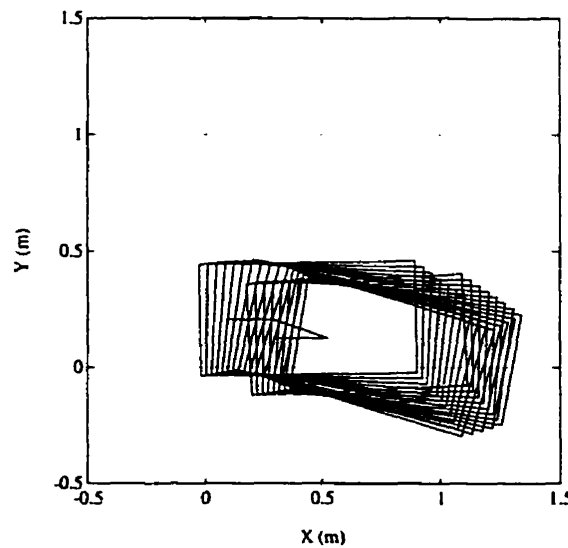


Figure 4.1 Résultat du planificateur

La figure 4.1 montre les résultats du planificateur pour un chemin simple. Dans cet exemple la fonction de coût à minimiser est le nombre de manœuvres, comme décrit dans [6]. La taille des cellules est de 2 cm, la configuration initiale (0.1, 0.2, 0.0) et la configuration finale (0.3, 0.1, 0.0). Sur une station de travail SUN SPARCStation II cet exemple requiert 10 minutes de calcul.

Cette méthode a comme avantage d'approximer un chemin optimal. Dans le cas du problème de stationnement le chemin souhaité est un chemin qui minimise le nombre de manoeuvres. Pour obtenir une bonne approximation du chemin il faut utilisé une discrétisation fine. Malheureusement, lorsqu'on utilise une discrétisation fine, le temps de calcul croît de façon astronomique. C'est ainsi que pour obtenir un résultat dans un temps raisonnable il faut utiliser une discrétisation grossière. Pour notre application les résultats obtenus à partir de discrétisation grossière ne sont pas appropriés. C'est-à-dire que plus la discrétisation est grossière plus la solution s'éloigne d'un chemin réalisable par le véhicule. Dans le cas du stationnement le chemin doit avoir une certaine précision pour éviter les collisions avec les obstacles.

4.3 Méthode de Laumond et Taix [11]

Cette technique est basée sur l'utilisation de chemins canoniques pour résoudre le problème de planification non-holonyme en présence d'obstacles. La méthode proposée par Laumond et Taix consiste à décomposer le problème en 2 étapes:

- Générer un chemin qui ignore les contraintes cinématiques du véhicule mais qui évite les obstacles.
- Générer un chemin réalisable, à l'aide des courbes de Reeds et Shepp, qui approxime le mieux possible le chemin holonyme.

4.3.1 Chemin holonome

Les planificateurs de chemin holonome assurent que tout mouvement arbitraire est possible à l'intérieur de l'espace des configurations. Le planificateur holonome développé est particulier au problème de stationnement. Il ne fait que générer un chemin holonome en forme de "L" entre la configuration initiale et la configuration finale et il vérifie s'il y a collision avec l'environnement. La figure 4.2 montre un exemple de chemin holonome généré par le planificateur.

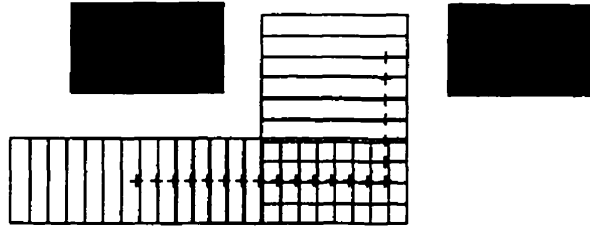


Figure 4.2 Chemin holonome

4.3.2 Chemin canonique

Dubins [12], Reeds et Shepp [11] ont démontré l'existence de chemins particuliers qui satisfont les contraintes cinématiques d'une automobile et donnent des trajectoires de longueur minimale. Ces chemins se composent de segments d'arcs de cercle ou de segments de droite. Les arcs de cercle ont un rayon égal au rayon de giration R_g minimal du véhicule. De plus les segments sont tangents entre eux et la position du chemin qui part du point initial ou qui part du point final est tangente à la direction imposée au point considéré.

Par la suite nous utiliserons les notations suivante:

- C_t est un arc de cercle de rayon R_q et l'indice t indique la longueur de l'arc.
On peut associer à une configuration donnée deux cercles tangents (un à droite et un à gauche). C est soit un arc de cercle gauche G soit un arc de cercle D .
- S_t est un segment de droite de longueur t .
- Un exposant $+$ ou $-$ indique la direction du déplacement (C^+, C^-, S^+, S^-).
- \parallel est un point de rebroussement.
- Un chemin de Reeds et Shepp est composé d'un ensemble de lettres $\{C, S, \parallel\}$.

Reeds et Shepp ont prouvé que les chemins de longueurs minimales sont constituées au maximum de cinq morceaux de type arc de cercle ou droite avec au plus deux points de rebroussement. De plus, ils montrent qu'on peut se limiter à étudier un nombre réduit de cas:

1. CSC
2. $C\parallel C\parallel C$
3. $C\parallel CC$
4. $CC_u\parallel C_uC$
5. $C\parallel C_uC_u\parallel C$
6. $C\parallel C_{\pi/2}SC$
7. $C\parallel C_{\pi/2}SC_{\pi/2}\parallel C$
8. $CC\parallel C$
9. $C\parallel SC_{\pi/2}C$

De ces 9 formules on peut déduire 48 chemins de Reeds et Shepp différents.

Pour caractériser complètement une courbe il faut connaître les 3 paramètres de

longueurs (t, u, v) . Nous présentons le calcul des trois paramètres pour le cas illustrés à la figure 4.3.

Les configurations initiale et finale sont respectivement $(0, 0, 0)$ et (x, y, θ) pour un chemin de type $G_t^+ D_u^- G_v^+$.

$$\begin{aligned} a &= d \cos \phi = x - \sin \theta R_g \\ b &= d \sin \phi = y + (\cos - 1) R_g \end{aligned} \tag{4.1}$$

$$\begin{aligned} \phi &= \arctan \frac{b}{a} \\ d &= \sqrt{a^2 + b^2} \end{aligned}$$

Donc:

$$\begin{aligned} t &= \left(\arccos \frac{d}{4R_g} + \frac{\pi}{2} - \phi \right)_{2\pi} \\ u &= \left(\pi - 2 \arccos \frac{d}{4R_g} \right)_{2\pi} \\ v &= (\theta - t - u)_{2\pi} \end{aligned} \tag{4.2}$$

La longueur de la courbe est $R_g(t + u + v)$.

On remarque que si d est supérieur à $4R_g$ alors la longueur est ∞ ce qui traduit le fait que ce type de courbe ne peut rejoindre les deux configurations.

Pour calculer la courbe de Reeds et Shepp entre deux configurations, il suffit de calculer la longueur associée aux 48 cas et de choisir la structure de courbe qui donne le chemin de longueur minimum.

4.3.3 Chemin non-holonyme

L'idée de la méthode de Taix et Laumond [10] consiste à générer un chemin réalisable, à l'aide des courbes de Reeds et Shepp, qui approxime le mieux possible le chemin holonyme. Voici les principales étapes pour transformer un chemin holonyme en un chemin non-holonyme.

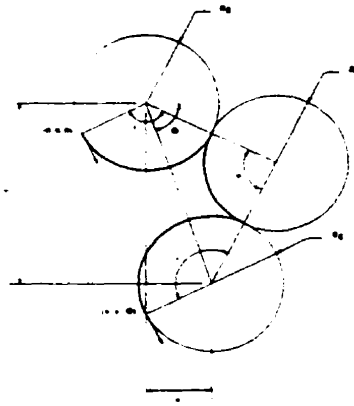


Figure 4.3 Exemple de chemin canonique $G_t^+ D_u^- G_v^+$

1. Relier les deux configurations extrêmes du chemin holonome à l'aide d'une courbe de Reeds et Shepp.
2. Tester les collisions
 - Si il y a collision, subdiviser le chemin holonome en deux chemins de longueur égales; reprendre à partir de 1 pour les deux chemins.
 - Sinon aller en 3.
3. Optimisation (Cette étape est répétée un nombre de fois arbitraire)
 - Générer de façon aléatoire l_1 et l_2 , où l_1 et l_2 sont plus petits que la longueur totale du chemin l_T .
 - Essayer de trouver un chemin plus court entre $[0, l_1]$, $[l_1, l_2]$ et $[l_2, l_T]$.

Cette méthode trouve toujours, s'il en existe un, un chemin réalisable. Mais on peut rien dire de ce chemin quant à son optimalité ou à ses propriétés. La figure 4.4 présente le chemin à différentes étapes de l'optimisation.

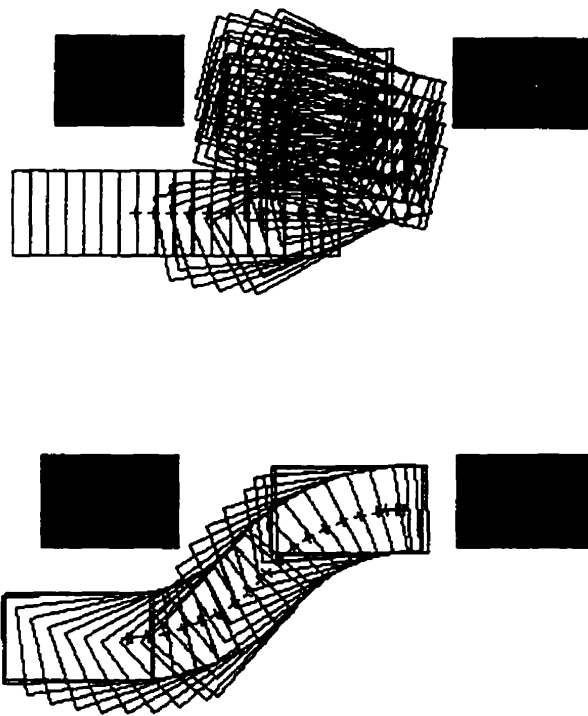


Figure 4.4 Méthode de Taix et Laumond

4.4 Conclusion

Dans ce chapitre nous avons décrit deux méthodes de planification de chemin. Les deux méthodes étudiées permettent la planification de chemin dans un environnement statique et préalablement connu. La première méthode, proposée par Barraquant et Latombe, est très générale mais ne donne pas des résultats satisfaisants (temps de résolution trop grand). Nous avons donc implanté la méthode de Taix et Laumond. Cette deuxième méthode donne un chemin construit de segment de droite et d'arc de cercle convenant mieux à notre application. Cette méthode proposée par Taix et Laumond est générale et s'applique peu importe l'environnement ou la forme du robot. Il est donc évident que pour résoudre le problème particulier du stationnement une méthode aussi puissante n'est pas

nécessaire. L'implantation que nous proposons est moins générale: le chemin holonome généré est en forme de "L" et est particulier au problème de stationnement. Un chemin non-holonome est par la suite généré à l'aide des courbes de Reeds et Shepp. Le chemin résultant est donc un chemin réalisable pour le véhicule. L'algorithme de planification résultant est donc spécialisé au problème de stationnement et est facile à implanter numériquement.

CHAPITRE 5

Description du système de vision

5.1 Introduction

Pour identifier un espace libre de stationnement ou pour se localiser par rapport à son environnement, le robot mobile doit être équipé d'un système de vision. Le capteur utilisé est une caméra CCD; cette caméra est jumellée à un numériseur pour produire une matrice d'intensité lumineuse. Le véhicule, muni d'une seule caméra CCD, se déplace par rapport à la scène et capte une séquence d'images monoculaires. Cette séquence d'images est traitée pour obtenir de l'information sous forme d'une carte de profondeur dense. Dans ce chapitre nous allons décrire les principales étapes de la méthode proposée par Weng [13], [15], [16], [17] et [18] (figure 5.1). Tout d'abord il faut identifier les paramètres de la caméra (section 5.2). Par la suite une stratégie d'appariement (section 5.3), utilisant la phase du signal image par transformée de Fourier, procure une carte de disparité de densité élevée pour chaque paire d'images consécutives. Finalement, une procédure non-linéaire d'estimation de la structure et du mouvement génère des cartes locales de profondeur à partir de la carte de disparité (section 5.4).

5.2 Calibration des paramètres de la caméra

La calibration précise de la caméra est cruciale pour les applications qui nécessitent des mesures quantitatives comme la mesure de la profondeur. Un des

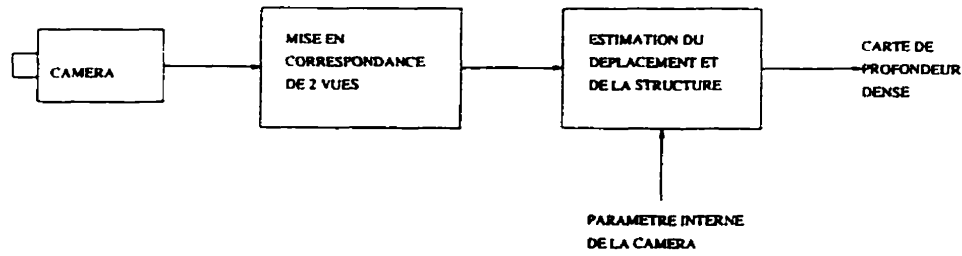


Figure 5.1 Système de vision

aspects de la calibration est l'estimation des paramètres internes de la caméra. Ces paramètres déterminent comment un point dans l'espace sera projeté sur le plan image de la caméra. L'estimation de la relation géométrique entre la caméra et la scène est aussi un des aspects importants de la calibration. Les paramètres qui caractérisent cette relation s'appellent les paramètres externes.

Le modèle considéré pour la procédure de calibration est le modèle "pin-hole". Soit (x_m, y_m, z_m) les coordonnées d'un point visible dans le référentiel monde et (x_c, y_c, z_c) représente les coordonnées du même point mais dans le référentiel de la caméra. L'origine du référentiel de la caméra coïncide avec le centre optique de la caméra (figure 5.2) et l'axe Z_c coïncide avec l'axe optique. La relation entre le référentiel monde et le référentiel de la caméra est donné par:

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = R \begin{bmatrix} x_m \\ y_m \\ z_m \end{bmatrix} + T \quad (5.1)$$

où $R = [r_{i,j}]$ est la matrice de rotation,

$T = (t_1, t_2, t_3)^T$ est la vecteur translation.

Le plan image, défini par le référentiel O' , est supposé parallèle au plan (x_c, y_c) et la distance f entre le plan image et l'origine du référentiel caméra est la distance focale. On peut exprimer le point P dans le plan image par les équations:

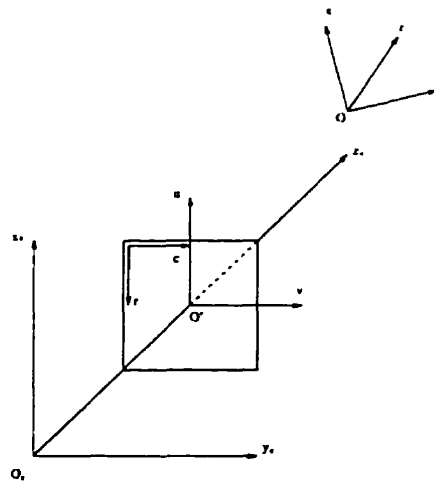


Figure 5.2 Système de coordonnées de la caméra

$$\begin{aligned} u &= f_u \frac{x_c}{z_c} \\ v &= f_v \frac{y_c}{z_c} \end{aligned} \quad (5.2)$$

où u et v sont les coordonnées du point dans le référentiel O' . Si r et c représentent la position du pixel correspondant à u et v dans l'image discrétisée alors on peut écrire que:

$$\begin{aligned} r - r_o &= -u \\ c - c_o &= v \end{aligned} \quad (5.3)$$

où (r_o, c_o) est le centre de l'image en pixel. En combinant (5.1), (5.2) et (5.3) on obtient l'équation reliant les paramètres internes (f_u, f_v, r_o, c_o) et externes de la

caméra $(r_{i,j}, t_i)$:

$$\begin{aligned} u &= \frac{r - r_o}{f_u} = \frac{r_{11}x_m + r_{12}y_m + r_{13}z_m + t_1}{r_{31}x_m + r_{32}y_m + r_{33}z_m + t_3} \\ v &= \frac{c - c_o}{f_v} = \frac{r_{21}x_m + r_{22}y_m + r_{23}z_m + t_2}{r_{31}x_m + r_{32}y_m + r_{33}z_m + t_3} \end{aligned} \quad (5.4)$$

où f_u est la distance focale selon l'axe u et f_v la distance focale selon l'axe v .

Le problème de la calibration consiste à résoudre un système d'équations non-linéaires. Le système d'équations est construit en écrivant les équations 5.4 pour un ensemble de points de contrôle. Pour chaque point de contrôle les coordonnées du point (x_m, y_m, z_m) sont connues puisque l'on utilise une scène connue et calibrée. Les coordonnées image (r, c) sont aussi connues puisque se sont les coordonnées des points dans l'image. La scène et le montage pour la calibration sont représentés à la figure 5.3.

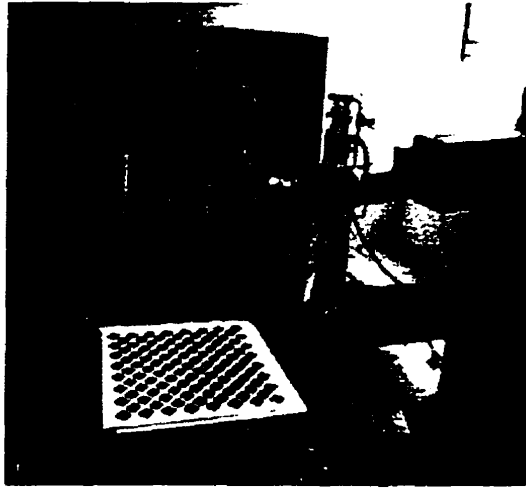


Figure 5.3 Montage pour la calibration

Pour résoudre le système, une méthode en deux étapes est implantée. La première étape consiste à estimer les paramètres $r_{i,j}, t_j, f_u, f_v, r_o, c_o$ en utilisant une procédure de minimisation de l'erreur quadratique moyenne pour l'expression:

$$F = \sum_i F_a^2(i) + F_b^2(i) \quad (5.5)$$

où:

$$\begin{aligned} F_a(i) &= (r - r_0)(r_{3,1}x + r_{3,2}y + r_{3,3}z + t_3) - f_u(r_{1,1}x + r_{1,2}y + r_{1,3}z + t_1) \\ F_b(i) &= (c - c_0)(r_{3,1}x + r_{3,2}y + r_{3,3}z + t_3) - f_v(r_{2,1}x + r_{2,2}y + r_{2,3}z + t_2) \end{aligned} \quad (5.6)$$

La seconde étape utilise une procédure itérative non-linéaire pour estimer la distorsion incluse dans les lentilles.

Les paramètres internes d'une caméra Pulnix avec une lentille de 8.5mm sont:

r_o	Le centre de l'image selon l'axe u	241 pixels
c_o	Le centre de l'image selon l'axe v	252 pixels
f_u	La distance focale selon l'axe u	-637 pixels
f_v	La distance focale selon l'axe v	525 pixels

5.3 Appariement

Le calcul de la structure 3D à partir d'images stéréoscopiques ou à partir d'une séquence d'images requiert que l'on établisse la correspondance entre les points des images. L'objectif de l'appariement est donc d'établir la correspondance pour une primitive sélectionnée de façon à ce que la correspondance des primitives proviennent du même élément dans la scène. La primitive utilisée dans la technique d'appariement est la phase de la transformée de Fourier à fenêtre (PTF). Cette primitive a pour avantage d'être disponible partout dans l'image ce qui permet d'obtenir des cartes de disparité dense contrairement à des primitives comme les arrêtes et les coins qui ont une densités faible.

La structure de calcul et le flux de données, utilisés dans cette technique de mise en correspondance à plusieurs niveaux, est représenté à la figure 5.4. Chaque niveau de résolution correspond à une taille de fenêtre pour la PTF. Cette structure de calcul multirésolution a pour avantage de diminuer le temps d'appariement.

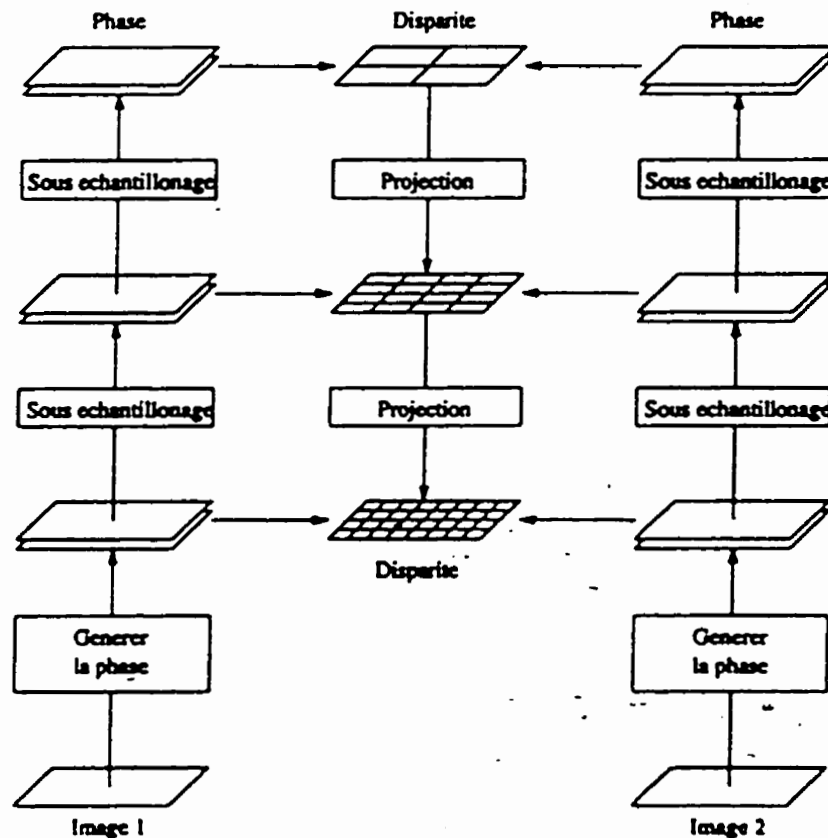
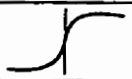
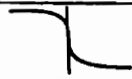
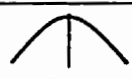
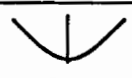
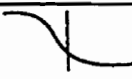
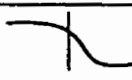
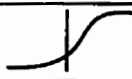
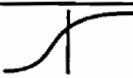


Figure 5.4 Structure de calcul

La phase d'un signal contient une information riche concernant la forme du signal. Le tableau 5.1 présente les différentes structures dans le signal image que permet d'identifier la PTF.

Ainsi, dans l'hypothèse où la forme de la surface d'intensité est préservée, ce qui est en général vrai dans le cas où les deux points de vue ne sont pas très différents l'un de l'autre, la phase représente un attribut fiable pour l'appariement. En général, les vecteurs de déplacement ne sont pas constants dans l'image.

Tableau 5.1 Structure dans le signal image

Nom	Forme d'onde	Phase
Les passages par zéro montants		$\frac{\pi}{2}$
Les passages par zéro descendants		$-\frac{\pi}{2}$
Les pics positifs		π
Les pics négatifs		0
Les pentes concaves descendantes		$\frac{\pi}{4}$
Les pentes convexes descendantes		$\frac{3\pi}{4}$
Les pentes concaves montantes		$-\frac{\pi}{4}$
Les pentes convexes montantes		$\frac{3\pi}{4}$

Cependant, les vecteurs de déplacement vus à partir d'une fenêtre de taille faible peuvent être assimilés à une translation constante. Ainsi, on définit une fenêtre rectangulaire avec le paramètre M comme suit:

$$w_M(u) = \begin{cases} 1 & \text{si } |u| \leq M/2 \\ 0 & \text{autrement.} \end{cases} \quad (5.7)$$

Le noyau de la transformation fenêtrée de Fourier est alors:

$$h(u) = w_M(u)e^{j2\pi\xi u}. \quad (5.8)$$

La transformation de Fourier d'un signal à une dimension $I(u)$ est exprimée comme suit:

$$g(u) = h(u) * I(u) = \int_{-\infty}^{\infty} w_M(t)I(u-t)e^{j2\pi\xi t} dt \quad (5.9)$$

$$= \int_{-M/2}^{+M/2} I(u-t) e^{j2\pi \xi t} dt \stackrel{\text{def}}{=} Re(u) + j Im(u) \quad (5.10)$$

où $Re(u)$ représente la composante réelle de g et $Im(u)$ sa composante imaginaire. La phase de la PTF est définie comme:

$$\phi(u) = \arg[g(u)]. \quad (5.11)$$

Soit $\phi_1(\mathbf{u})$ la phase de la transformée fenêtrée de Fourier associé à l'image I_1 et $\phi_2(\mathbf{u})$ la phase de l'image I_2 , un point \mathbf{u} dans la première image est apparié avec le point $\mathbf{u} + \mathbf{d}$ dans la deuxième image si la différence de phase r_1 est petite.

$$r_1 = \phi_2(\mathbf{u} + \mathbf{d}(\mathbf{u})) - \phi_1(\mathbf{u}) \quad (5.12)$$

De plus, on impose la contrainte de douceur au champ de déplacement. Ceci revient à introduire le terme suivant:

$$r_2 = \|\mathbf{d}(\mathbf{u}) - \bar{\mathbf{d}}(\mathbf{u})\| \quad (5.13)$$

où $\bar{\mathbf{d}}$ représente le déplacement moyen calculé à l'intérieur d'une région autour de \mathbf{u} . Dans les régions uniformes où la phase n'a pas de signification, la disparité est interpolée automatiquement.

Le résultat de cet algorithme est une carte de disparité dense. C'est-à-dire une carte représentant le déplacement d'un pixel d'une image par rapport à une autre.

5.4 Estimation de la structure

Le calcul de la structure 3D consiste à déterminer, pour chaque paire de points appariés, les coordonnées (x, y, z) du point physique correspondant dans la scène.

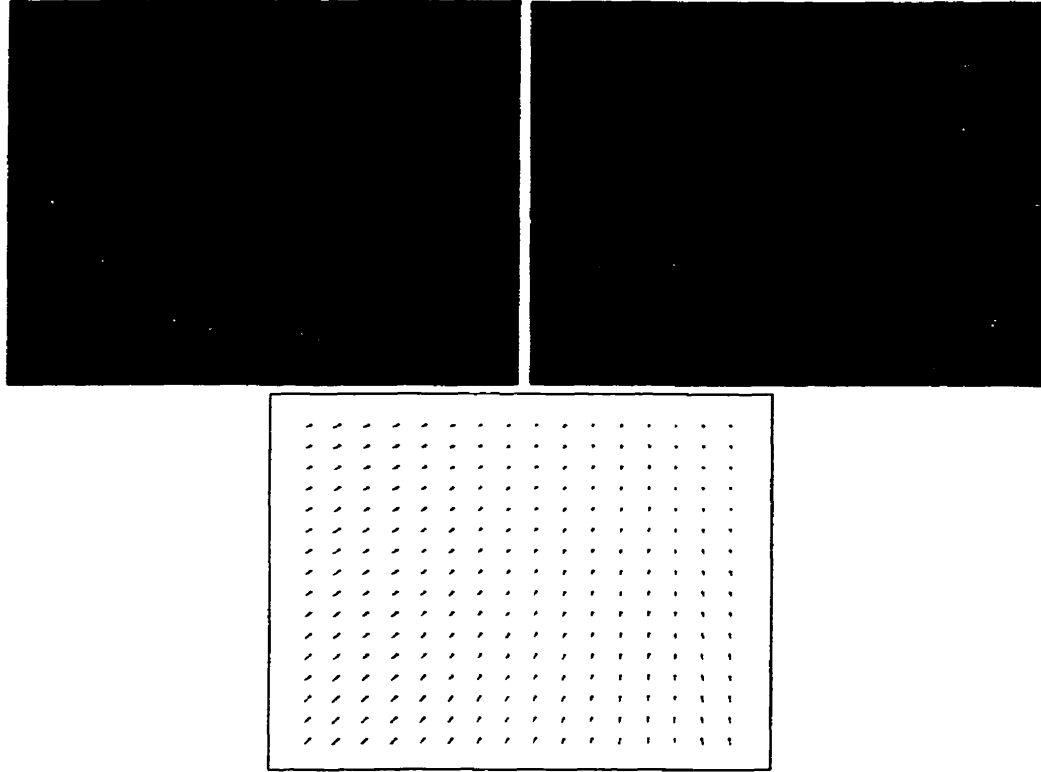


Figure 5.5 Résultat de l'appariement

Le mouvement de la caméra entre deux prises d'images est nécessaire pour estimer la structure. Le mouvement général d'une structure 3D rigide entre deux instants consécutif $k-1$ et k peut être défini par le vecteur de translation $T_{k-1,k}$ et la matrice de rotation $R_{k-1,k}$. La relation à l'instant k entre la position $X_{k-1} = (x_{k-1}, y_{k-1}, z_{k-1})^T$ d'un point particulier de l'image à l'instant $k-1$ et sa position $X_k = (x_k, y_k, z_k)^T$ à l'instant k est donnée par:

$$X_k = R_{k-1,k} X_{k-1} + T_{k-1,k}. \quad (5.14)$$

La projection du point X dans le référentiel du plan image à l'instant k est:

$$\begin{aligned} u &= \frac{x_{k-1}}{z_{k-1}} \\ v &= \frac{y_{k-1}}{z_{k-1}} \end{aligned} \quad (5.15)$$

La projection du point X_k dans le référentiel du plan image à l'instant k est:

$$\begin{aligned} u' &= \frac{r_{11}x_{k-1} + r_{12}y_{k-1} + r_{13}z_{k-1} + t_1}{r_{31}x_{k-1} + r_{32}y_{k-1} + r_{33}z_{k-1} + t_3} \\ v' &= \frac{r_{21}x_{k-1} + r_{22}y_{k-1} + r_{23}z_{k-1} + t_2}{r_{31}x_{k-1} + r_{32}y_{k-1} + r_{33}z_{k-1} + t_3} \end{aligned} \quad (5.16)$$

Le problème d'estimation du mouvement, comme le problème de calibration, consiste à résoudre un système d'équations non-linéaires. Le système d'équation est construit en écrivant les équations 5.15, 5.16 pour un ensemble de points d'ancrages. Les points d'ancrages sont choisis de façon aléatoire à travers la carte de disparité. Pour résoudre le système d'équation une méthode semblable à celle utilisée lors de la calibration est implantée.

Le calcul de la structure 3D consiste à déterminer, pour chaque paire de points appariés, les coordonnées (x, y, z) du point physique correspondant dans la scène. Soit (P_1, P_2) le couple de points appariés. Parce que le processus d'appariement génère du bruit, les deux droites (O_1P_1) et (O_2P_2) ne se coupent pas en général. Par conséquent, une méthode exacte de triangulation ne peut être appliquée, et une approximation est alors nécessaire.

Soit P'_1 un point sur la droite (O_1P_1) et P'_2 sur la droite (O_2P_2) , le point le plus plausible pour la solution est le point qui est au milieu du plus court segment $P'_1P'_2$.

5.5 Conclusion

Dans ce chapitre nous avons décrit la technique de vision qui nous permet à partir d'une séquence d'images d'obtenir de l'information télémétrique. Cette technique

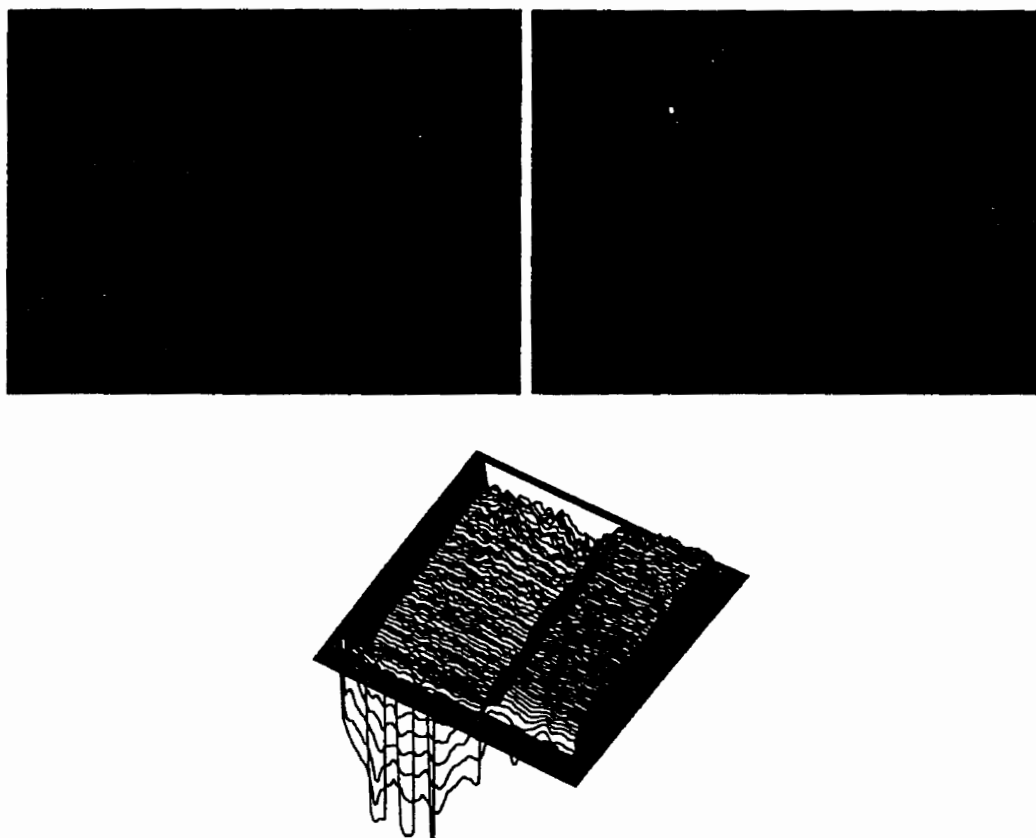


Figure 5.6 Résultat de l'estimation de la structure

comporte trois étapes principales:

- La calibration;
- L'appariement;
- L'estimation de la structure et du mouvement.

L'appariement est l'étape crucial du système de vision, ainsi la qualité de cette appariement influence directement la qualité de la carte de profondeur obtenue. La qualité de l'appariement varie beaucoup en fonction du contenu

de l'image. Pour s'assurer d'un bon appariement il faut des images texturées. Cette étape est aussi très couteuse en temps de calcul.

A partir de ces cartes locales de profondeur nous allons construire une carte globale représentant la scène afin d'identifier l'espace de stationnement. Au chapitre suivant nous démontrons que les cartes locales de profondeur obtenue grâce à cette technique sont d'une précision suffisante pour permettre l'identification d'un espace libre de stationnement.

CHAPITRE 6

Identification de l'espace de stationnement

6.1 Introduction

Pour identifier l'espace de stationnement le véhicule doit obtenir des données télémétriques sur son environnement. Pour ce faire pendant que le véhicule se déplace parallèlement à la scène, la caméra est pointée à 90° par rapport à la direction du mouvement et capte une séquence d'images de l'environnement global.

A partir de cette séquence, une représentation tridimensionnelle du profil de stationnement est calculée. Les cartes locales sont intégrées en un profil global du stationnement. Pour identifier l'espace libre, un modèle est agencé au profil globale.

Lors de l'exécution des manoeuvres de stationnement, des erreurs de déplacements, causées principalement par le glissement des roues font dévier le véhicule de sa trajectoire planifiée. La capture d'information visuelle locale permet de connaître la position réelle du véhicule et de corriger cette déviation.

6.2 Construction de la carte globale

Pour construire la carte globale, le véhicule doit se déplacer parallèlement à la scène (figure 6.1), la caméra est pointée à 90° par rapport à la direction du mouvement et capte une séquence d'images de l'environnement global. Lors de

la prise de chaque image, la position du véhicule est enregistrée. Par le procédé décrit au chapitre précédent, les cartes de profondeurs locales sont générées.

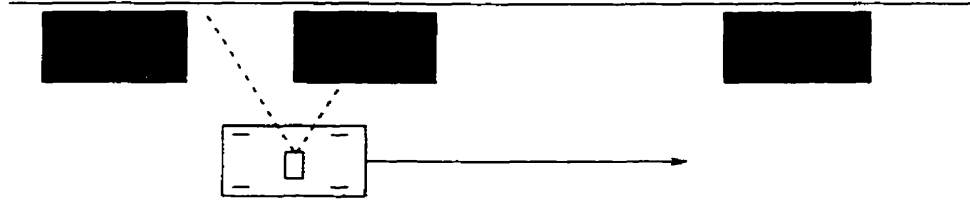


Figure 6.1 Scène

La mise à l'échelle des cartes de profondeur se fait en multipliant les coordonnées (x, y, z) d'un point par la norme du vecteur translation entre les deux images. Le vecteur translation est fourni par l'odométrie du véhicule. Suite à cette mise à l'échelle, on obtient des cartes locales contenant les dimensions réelles de la scène.

Les coordonnées (x, y, z) des points dans les cartes locales sont exprimés par rapport au référentiel de la caméra (figure 6.2). Il faut donc pour obtenir ces coordonnées dans le référentiel du véhicule effectuer la transformation suivante:

$$\begin{bmatrix} x_r \\ y_r \\ z_r \end{bmatrix} = R_r^c \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} + T_r^c \quad (6.1)$$

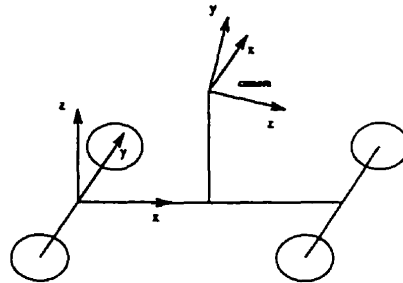


Figure 6.2 Référentiel de la caméra

L'intégration des cartes locales se fait à partir de l'information de position de la prise des images. On utilise comme référentiel global, la position du véhicule lors de la prise de la première image. Pour toutes les autres cartes locales, on effectue la transformation suivante pour exprimer les coordonnées des points des cartes locales dans le référentiel global:

$$\begin{bmatrix} x_m \\ y_m \\ z_m \end{bmatrix} = R_g^r \begin{bmatrix} x_r \\ y_r \\ z_r \end{bmatrix} + T_g^r \quad (6.2)$$

où R_g^r et T_g^r dépendent de la position du véhicule. A la figure 6.3 on voit quelques images composant une séquence, les cartes locales associées aux images et la carte globale.

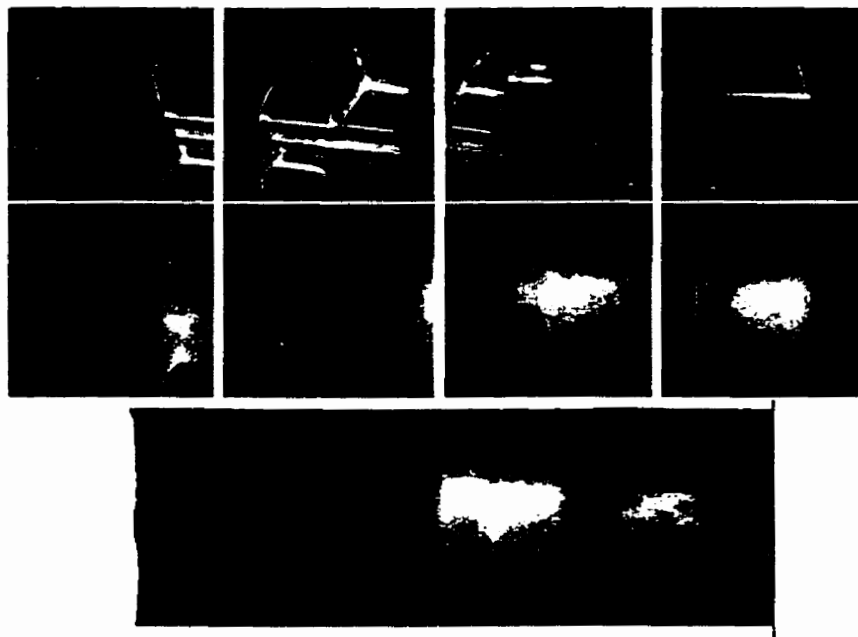


Figure 6.3 Carte globale

6.3 Identification de l'espace de stationnement

L'analyse de la carte globale, qui permet d'identifier l'espace de stationnement, comprend 3 étapes (figure 6.4):

- Filtrage de la carte globale;
- Création du profil de stationnement;
- Agencement du modèle au profil de stationnement.

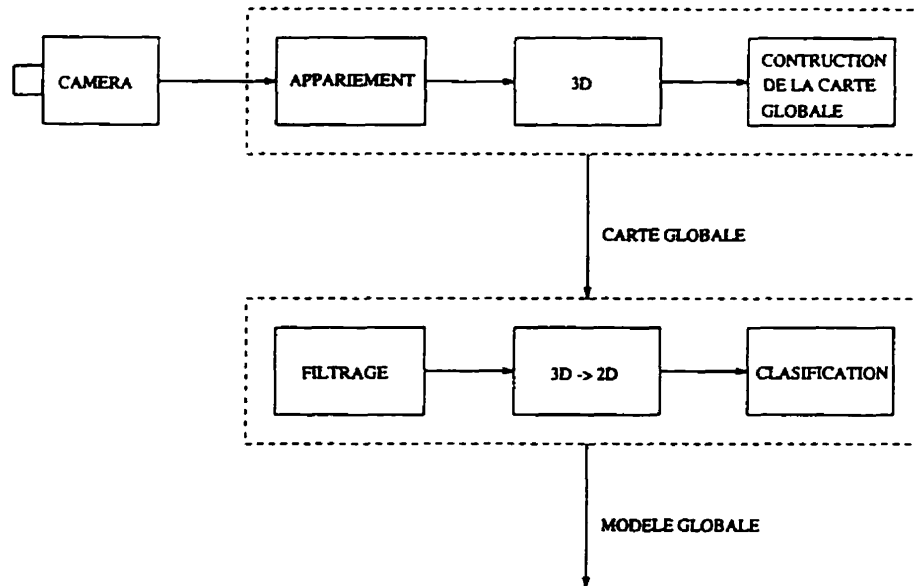


Figure 6.4 Identification de l'espace de stationnement

La carte globale est entachée d'une importante quantité de bruit, un filtrage spatial est donc nécessaire pour extraire le profil de stationnement. Chaque point de la carte de profondeur filtrée est obtenue en calculant la moyenne des points de son voisinage.

Une fois la carte filtrée, on extrait le profil de stationnement. L'extraction se fait en projetant les points P_{ij} de la carte globale sur le plan XY suivant l'équation:

$$P_i = \frac{1}{N} \sum_{j=0}^N P_{ij} \quad (6.3)$$

où i est l'indice de la grille suivant l'axe x et j suivant z .

Le vecteur obtenu représente donc le profil et les valeurs P_i de ce vecteur sont les distances suivant y le long de l'axe x . A la figure 6.5 un exemple de profil est présenté. Le tableau 6.1 donne la précision du profil de stationnement suivant x et y .

Tableau 6.1 Précision du profil

Axe	Erreur
x	0.021m
y	0.032m

L'espace de stationnement est identifier à partir du profil. Nous allons identifier l'arrière plan et le coté des voitures. Une méthode d'aggrégation semble être un bon moyen pour déterminer si un point du profil appartient à l'arrière plan ou au coté d'une voiture. La méthode adoptée est l'algorithme d'aggrégation flou développé par Dunn [20]. L'ensemble des données $\chi = X_1, X_2, \dots, X_n$ est associé à un ensemble de N objets. Chaque élément X_i est un vecteur de caractéristiques de dimension P associé à un objet particulier de l'ensemble. Dans notre cas le vecteur de caractéristique X_i est de dimension $P = 1$ et a comme valeur un point P_i du profil de stationnement.

L'algorithme tente de partitionner l'ensemble des données χ en C sous-ensembles qui décrivent la structure interne de l'ensemble de données. Le résultat de la partition est représenté par une matrice U (matrice de partition) de dimension $N \times C$, qui se compose des éléments u_{ij} . Le coefficient u_{ij} mesure l'appartenance de l'élément j au sous ensemble i ; u_{ij} doit satisfaire les propriétés

suivantes:

$$\begin{aligned} u_{ij} &\in [0, 1] \\ \sum_{j=1}^N u_{ij} &> 0 \forall i \\ \sum_{i=1}^C u_{ij} &= 1 \forall j \end{aligned} \quad (6.4)$$

L'algorithme d'aggrégation flou utilise une optimisation itérative pour approximer le minimum de la fonction de coût suivante:

$$J_m(U, V) = \sum_{j=1}^N \sum_{i=1}^C (u_{ij})^m (d_{ij})^2 \quad (6.5)$$

où: U est la partition flou de χ ;

$V = V_1, \dots, V_C$ avec V_i désignant le centre de la classe i ;

$d_{ij} = \|V_i - V_j\|$;

$m \in [1, \text{inf}]$.

Les principales étapes de l'algorithme sont:

1. Calculer le centre de la classe $\{V_i^{(n)}\}$ de $U^{(n)}$:

$$v_{il} = \frac{\sum_{j=1}^N (u_{ij})^m x_{il}}{\sum_{j=1}^N (u_{ij})^m} \quad l = 1, 2, \dots, p \quad (6.6)$$

2. Mise à jour de U^n en calculant $U^{(n+1)}$: pour j de 1 à N

(a) Déterminer le contenu des ensembles suivant:

$$I_j = \{j \text{ telque } d_{ij} = 0\} \quad (6.7)$$

$$\bar{I}_j = \{1, 2, \dots, C\} - I_j \quad (6.8)$$

(b) Pour les données j calculer la nouvelle valeur d'appartenance

- si $I_j = 0$

$$u_{ij} = \frac{1}{\sum_{k=1}^C (d_{ij}/d_{kj})^{\frac{2}{(m-1)}}} \quad (6.9)$$

- sinon $u_{ij} = 0$ pour tout $i \in \bar{I}_j$ et $\sum_{i \in I_j} u_{ij} = 1$
- - si $\|U^{(n)} - U^{(n+1)}\| < \varepsilon$, fin.
- sinon $n = n + 1$, faire 1.

Une fois les points du profil classés, on identifie aisément un espace suffisamment grand. La figure 6.5 montre le résultat de la classification. Malgré la présence de points incorrects dans le profil, dû à des erreurs dans l'appariement, le résultat de l'aggrégation est bon.

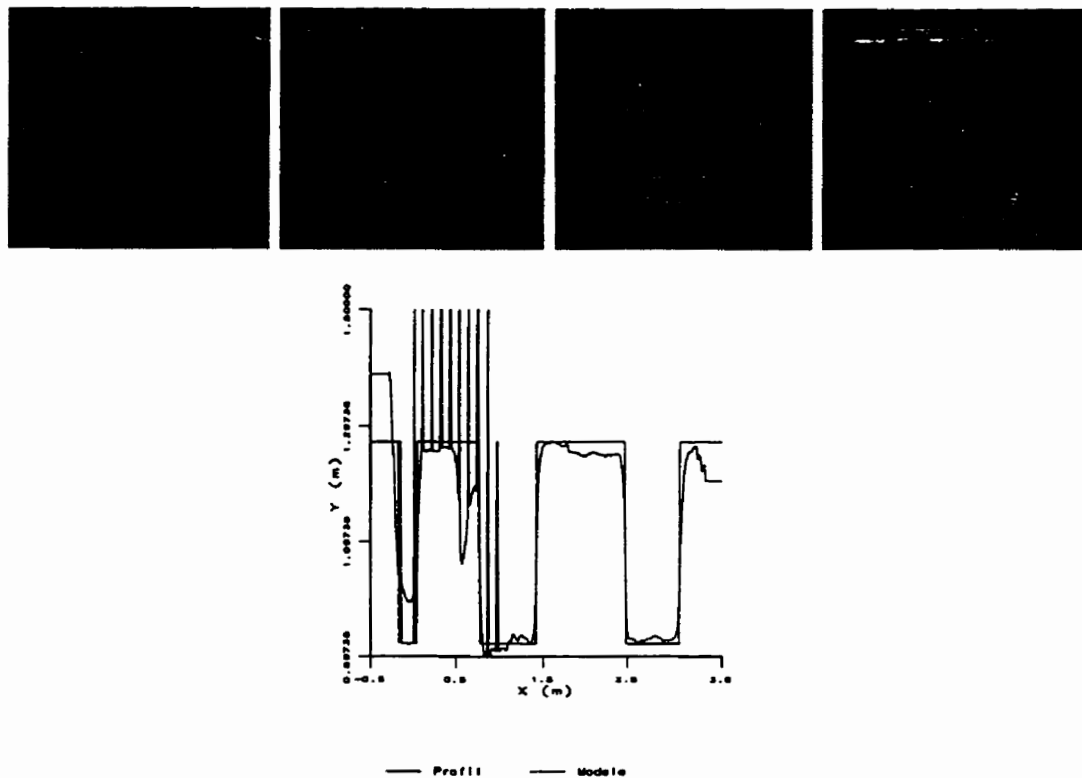


Figure 6.5 Exemple d'identification de l'espace de stationnement

6.4 Rétro action visuelle

La rétro-action visuelle permet, à partir d'une carte de profondeur locale, de localiser le véhicule par rapport au modèle global lors de la manoeuvre de stationnement. L'analyse de la carte locale s'effectue en trois étapes (figure 6.6):

- Filtrage de la carte locale;
- Création du profil local;
- Maximisation de la corrélation entre le modèle global et le profil local.

Les méthodes utilisées pour le filtrage et la création du profil sont les mêmes que pour l'identification de l'espace de stationnement. Pour localiser le véhicule lors de la manoeuvre de stationnement nous allons mettre en correspondance un profil local et le modèle global. Connaissant la position de la carte globale par rapport au référentiel global et connaissant le profil local par rapport au référentiel de la caméra on obtient, après la mise en correspondance du profil local et du modèle global, la position du véhicule dans le référentiel global. La corrélation semble l'outil idéal pour quantifier la qualité de la correspondance. La mise en correspondance revient donc à trouver la position du profil qui maximise la corrélation. Pour simplifier le problème nous traitons les deux profils comme deux signaux à une dimension. P_r étant le vecteur qui représente le profil local et M_g le vecteur du modèle global. La corrélation entre ces deux profils est donnée par:

$$corr = \frac{E(P_r M_g) - E(P_r)E(M_g)}{\sqrt{(E(P_r^2) - E(P_r)^2)(E(M_g^2) - E(M_g)^2)}} \quad (6.10)$$

La figure 6.7 montre le modèle local ainsi que la position du profil local qui maximise la corrélation et la fonction de corrélation pour ce profil local. La

précision du profil local est la même que celle de la carte globale (tableau 6.1).

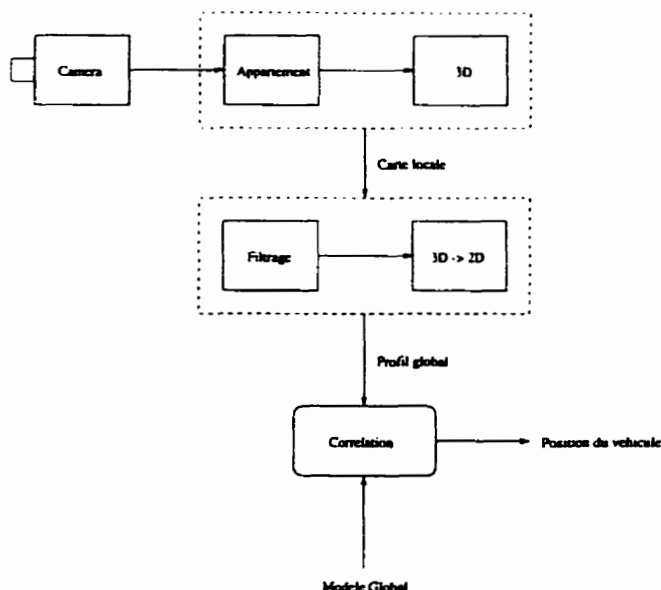


Figure 6.6 Étape de la relocalisation

6.5 Conclusion

Dans ce chapitre nous avons mis en place les derniers éléments qui permettent d'effectuer le stationnement. C'est ainsi que nous avons décrit la méthode pour identifier l'espace de stationnement et la méthode pour relocaliser le véhicule lors de la manoeuvre de stationnement. Ces deux méthodes permettent de localiser le véhicule dans son environnement. La méthode proposée pour identifier l'espace libre de stationnement permet de localiser avec précision la position de l'espace libre ainsi que la position des véhicules dans la rangée. Pour la relocalisation du véhicule lors de la manoeuvre de stationnement, la méthode proposée est elle aussi précise. Mais dans le cas particulier du stationnement l'erreur de localisation de la méthode par la vision est du même ordre de grandeur que l'erreur de la localisation par odométrie. L'erreur d'odométrie que nous avons évalué au chapitre 2 est

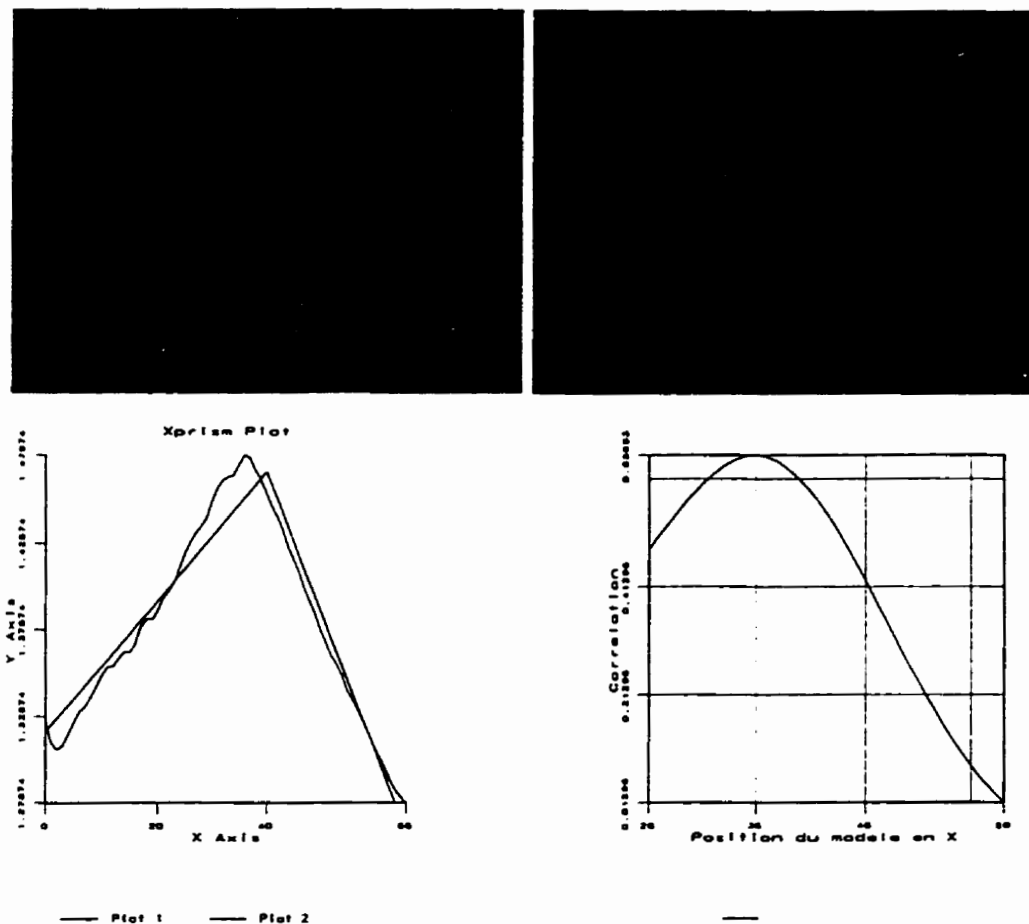


Figure 6.7 Profil local et modèle

de l'ordre de 0.05m pour un demi-cercle de 0.7m de rayon et l'erreur dans la construction des profils est de l'ordre de 0.04m (tableau 6.1).

CHAPITRE 7

Conclusion

Dans ce travail, nous avons conçu, implanté et intégré un système de stationnement automatisé qui comprend les modules suivants:

- Identification de l'espace de stationnement.
- Planification de chemin.
- Suivi de chemin.

Pour nous permettre d'effectuer les nombreux essais expérimentaux qu'implique la réalisation d'un tel système nous avons développé un véhicule de petite taille. Les équations odométriques et le modèle dynamique de ce robot ont été développées. Des essais expérimentaux ont démontré la validité des équations odométriques et du modèle dynamique.

La stratégie de contrôle a été développée de façon à effectuer le suivi de chemin géométrique. Afin de concevoir des contrôleurs simples pour le suivi de chemin, le modèle dynamique a été linéarisé et découplé de façon à obtenir deux sous-systèmes: un qui régit le mouvement en translation et l'autre le mouvement en rotation. Les contrôleurs ainsi que l'algorithme de suivi de chemin ont été implantés et les essais expérimentaux ont montré leur efficacité.

Pour la planification de chemin, nous nous sommes intéressés aux méthodes de planification de chemins non-holonomes pour des environnements avec des obstacles statiques. Plus particulièrement nous avons étudié la méthode de Barraquant et Latombe et la méthode de Taix et Laumond. La méthode de Barraquant et Latombe est très générale mais ne donne pas des résultats satisfaisants

pour le problème de stationnement. La méthode de Taix et Laumond donne un chemin composé de segments de droite et d'arcs de cercle convenant mieux à notre application.

Pour identifier l'espace libre de stationnement et pour se localiser par rapport à son environnement, le robot mobile a été équipé d'un système de vision. Ce système de vision permet d'obtenir l'information télémétrique sous forme de cartes de profondeur denses à partir d'une séquence d'images. L'information télémétrique est intégrée en un profil global du stationnement. Pour identifier l'espace libre, un modèle est agencé au profil global.

Lors de l'exécution des manoeuvres de stationnement, des erreurs de déplacement, causées principalement par le glissement des roues, sont corrigées par la capture d'informations visuelles locale qui permettent de connaître la position du véhicule.

Le système de stationnement autonome tel que décrit ne peut pas être implanté directement sur un véhicule réel. En effet, les contrôleurs ont été développés pour une voiture électrique et non pour une voiture à essence. De plus, le système de vision utilisé ne peut fonctionner en temps réel. Il faudrait donc un système qui fournisse l'information télémétrique plus rapidement comme par exemple, un télémètre laser. Dans le contexte d'un système autonome en présence d'humains il faudrait améliorer la sécurité. Entre autre, il faudrait résoudre le problème de détection et d'évitement d'obstacles.

BIBLIOGRAPHIE

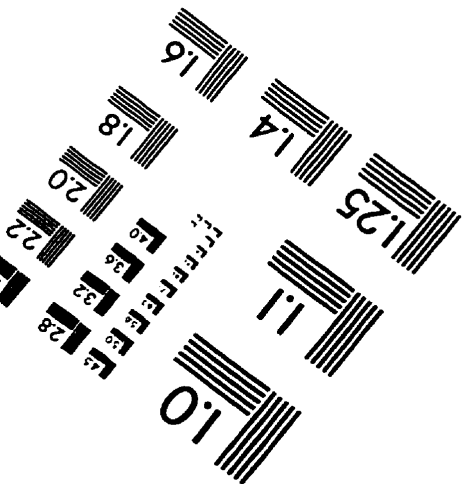
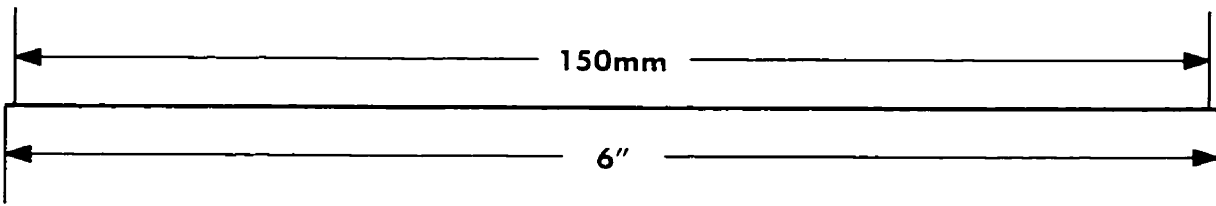
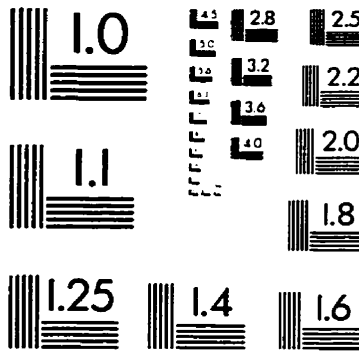
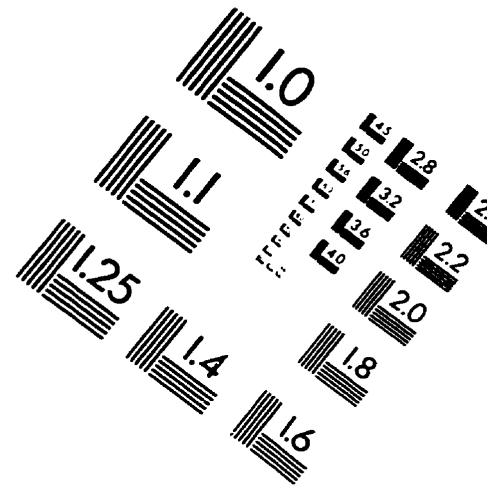
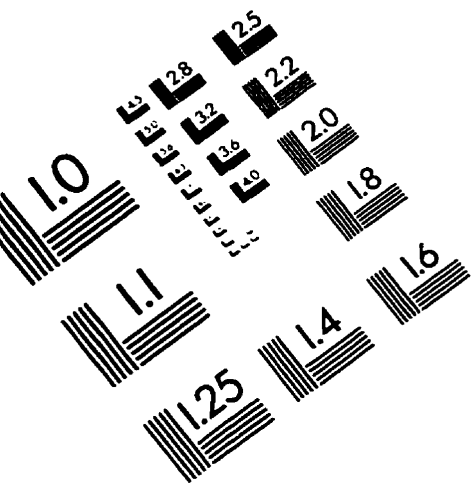
- [1] KILLOUGH S.M., PIN F.G., "Design of an Omnidirectional and Holonomic Wheeled Platform Prototype", Proceedings of the 1992 IEEE International Conference on Robotics and Automation, Nice, pp. 84-90.
- [2] GIRALT G., CHATILA R., VAISSET M., "An integrated Navigation and Motion Control System for Autonomous Multisensory Mobile Robots", Autonomous Robot Vehicles, edited by Cox I.J., Wilfong G.T., Springer-Verlag, pp. 420-443.
- [3] DE SANTIS R.M., HURTEAU R., PICHÉ A., "On the Motion Control of Electric AGVS", Second Workshop on Military Robotic Applications, Kingston, août 1989, pp. 306-315.
- [4] DE SANTIS R.M., "Path-Tracking for Car-Like and Tractor-Trailer-Like Robots", The International Journal of Robotics Research, Vol. 13, No. 6, décembre 1994, pp. 533-544.
- [5] DE SANTIS R.M., "Path-Tracking for a Mobile Wheeled Robot with a Differential Drive", Rapport technique, École Polytechnique, EPM/RT-94/11, mars 1994, 28p.
- [6] LATOMBE J.C., "Robot Motion Planning", Kluwer Academic Publisher, 1991, 651p.
- [7] JUNEAU L., "Étude d'un système de guidage autonome pour des environnements structurés", Mémoire de M.Sc.A., École Polytechnique, août 1993, 131p.

- [8] BARRAQUAND J., LATOMBE J.-C., "On Nonholonomic Mobile Robots and Optimal Maneuvring", Proceeding IEEE International Symposium on Intelligent Control, septembre 1989, Albany, pp. 340-347.
- [9] LAUMOND J.-P., "Controllability of a Multibody Mobile Robot", I.C.A.R. 91, juin 1991, Pisa, Italia.
- [10] TAIX M., "Planification de mouvement pour robot mobile non- holonome", Université Paul Sabatier, Toulouse, 1991, 126p.
- [11] REEDS J. A., SHEPP L. A., "Optimal Paths for a Car that Goes Both Foward and Backward", Pacific Journal of Mathematics, Vol. 145, 1990.
- [12] DUBINS L., "On Curves of Minimal Length with a Constraint on Average Curvature and with Prescribed Initial and Terminal Position and Tangent", American Journal of Mathematics, 1957, pp. 497-516.
- [13] CUI N., WENG J., COHEN P., "Recursive-Batch Estimation of Motion and Structure From Monocular Image Sequences", CVGIP: Image Understanding, Vol. 59, Num 2, mars 1994, pp. 154-170.
- [14] GILL S., HURTEAU R., COHEN P., PIEDBOEUF J.-C., "Conception d'un mini-véhicule pour l'étude de système de guidage autonome", DND Workshop on Advanced Technology in Knowledge Based Systems and Robotics, novembre 1993, pp. 173-177.
- [15] WENG J., COHEN P., HERNIOU M., "Acurate Calibration of Stereo Cameras Based on Modeling and Correction of Radial and Tangential Distorsion", IEEE Transaction on Pattern Analysis and Machine Intelligence, octobre 1992, Vol. 14, No. 10, pp. 965-980.

- [16] WENG J., COHEN P., REBIBO N., "A Computational Study of Motion and Structure Estimation from Point Matches in Stereo Image Sequences", IEEE Transaction on Robotics and Automation, juin 1992, Vol. 8, Num. 3 pp. 362-382.
- [17] AUDETTE M., CUI N., COHEN P., WENG J., "Estimation of Ground Structure and Aircraft Motion from Aerial Image Sequences", Canadian Conf. on Electrical and Computer Engineering, Quebec, septembre 1991.
- [18] GUILLEMIN Y., NGUYEN H. H., COHEN P., "Automated Aerial Cartography from Extended Sequences", Scandinavian Conference on Image Analysis, Tromso, Norway, mai 1993, 20p.
- [19] BEZDEK J. C., "Pattern Recognition with Fuzzy Objective Function Algorithms", Plenum, New-York, 1981.
- [20] DUNN J. C., "A Fuzzy Relative ISODATA Process and its Use in Detecting Compact, Well Separated Clusters", J. Cybernetics 3, 1974, pp. 32-57.
- [21] KATZ P., "Digital Control Using Microprocessors", Prentice-Hall international, 1981, 293 p.
- [22] FRANKLIN G. F., POWELL J. D., WORKMAN M. L., "Digital Control of Dynamic Systems", Addison Wesley, Second Edition, 1989, 837p.
- [23] PHILLIPS C. L., HARBOR R. D., "Feedback Control Systems", Prentice Hall, 1988, 626p.
- [24] NGUYEN D. H., WIDROW B., "Neural Networks for Self-Learning Control System", IEEE Control Systems Magazine, pp. 18-23, avril 1990.

- [25] DRISS A., RODRIGUES V., COHEN P., "Parking a Vision-Guided Car-Type Vehicule". IEEE Conference on Control Appl., Glasgow, UK, août 1994.
- [26] CHI-CHENG J., NAU-CHING W. "Training a Fuzzy Controler to Backup an Autonomous Vehicule",

IMAGE EVALUATION TEST TARGET (QA-3)



APPLIED IMAGE, Inc.
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989

© 1993, Applied Image, Inc., All Rights Reserved

